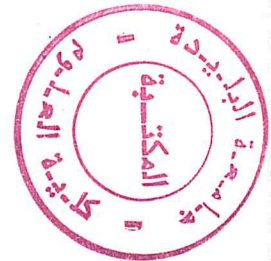


République Algérienne Démocratique et Populaire.  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Saad Dahlab, Blida  
USDB.

Faculté des sciences.  
Département informatique.



Mémoire pour l'obtention  
d'un diplôme d'ingénieur d'état en informatique.

Option : Système d'Information

Sujet :

Conception et réalisation d'une base  
de donnée Multi-Niveaux médicale

Présenté par: *Benrekja Nabila*  
*Azezda Nabihia*

Promotrice: *Boustia Narhimene*

Soutenu le: date soutenance, devant le jury composé de :

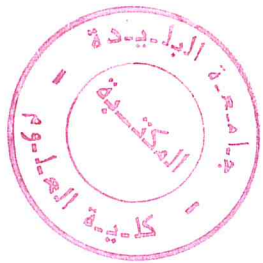
Nom. président du jury : M<sup>me</sup> *Bensttiti .S*

Nom examinateur 1 : M<sup>lle</sup> *FARAH*

Nom examinateur 2 : M<sup>lle</sup> *AOUSAT*

Promotion N° 03 / 2004 - 2005

MIG-004-69-1





# Remerciements

*Quelques lignes ne pourront jamais exprimer la reconnaissance que nous éprouvons envers tous ceux qui, de près ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leurs amitiés à l'aboutissement de ce travail.*

*Nos vifs remerciements accompagnés de toute notre gratitude vont tout d'abord à notre promotrice M<sup>lle</sup> BOUSTIA NARHIMENE, pour les conseils qu'elle n'a cessé de nous prodiguer et surtout pour la confiance qu'elle nous a accordée pour la réalisation de ce projet.*

*Nous tenons à remercier notre chef du département d'informatique  
M<sup>me</sup> BENSTITI SOUAD.*

*Nous tenons à remercier tous les membres du jury qui ont accepté de jury ce  
Travail.*

*Notre reconnaissance va particulièrement à tous nos enseignants du département  
d'informatique.*

## *SOMMAIRE*

### **Chapitre I : Généralité sur les bases de données et SGBD**

Introduction.....	1
1. Définition d'une base de donnée.....	1
2. La gestion des bases de données.....	2
2.1 les architectures d'un SGBD .....	3
2.1.1 L'architecture client – serveur.....	3
2.1.2 L'architecture repartis.....	4
2.2 Les caractéristiques du système de gestion de base de donnée.....	4
3. Le modèle relationnel de données.....	5
3.1 Les objectifs du modèle relationnel .....	5
3.2 Base de données relationnelles.....	5
4. Les extensions du modèle relationnel et les modèles objets.....	6
5. Modèle orientée objet .....	7
5.1 L'approche objet .....	7
5.2 Base de données orientées objet.....	8
5.3 SGBD orienté-objet.....	8
5.4 Objectifs des SGBDOO.....	8
6. Problématique de la sécurité.....	8
6.1 Contrôles d'accès discrétionnaires.....	9
6.2 Contrôles d'accès obligatoires .....	9
Conclusion .....	11

### **Chapitre II : la sécurité multi-niveau et représentation du modèle Multi-Vues**

Introduction.....	12
1. les approches de sécurité Multi-niveau .....	12
L'approche filtre .....	12
L'approche noyau de sécurité .....	12
L'approche sujet de confiance .....	13
L'approche répliqué .....	13

2.La confidentialité.....	13
3.Granularité de la classification .....	14
4.Gestion des leurres.....	16
5.Inférence nom autorisée d'information.....	17
6.Représentation du Modèle Multi-Vues .....	18
6.1 Définition.....	18
6.2 Exemple.....	19
Conclusion.....	24

### **Chapitre III : Méthode de conception et présentation de la méthode**

#### **OMT**

Introduction.....	25
1.Le rôle des méthodes.....	25
2.Les différentes classes de méthodes de conception .....	25
2.1 Les approches cartésiennes (première génération).....	26
2.2 Les approches systémiques (deuxième génération) .....	26
2.3 Les approches objets (troisième génération).....	27
3.Pourquoi l'OMT ? .....	28
4.Présentation de la méthode « OMT » (Object Modeling Technique) .....	28
4.1 Le modèle statique.....	28
4.2 Le modèle dynamique.....	29
4.2.1 Scénario.....	29
4.2.2 Diagramme de suivi d'événement.....	30
4.2.3 Diagramme d'état.....	31
4.3 Modèle fonctionnel.....	32
5.La démarche méthodologique .....	33
5.1 L'analyse.....	34
5.2 La Conception du système.....	34
5.3 La conception des objets.....	35
Conclusion.....	35

### **Chapitre IV : Analyse et Conception**

Introduction .....	36
1. Analyse .....	36
1.1 Construction du modèle objet.....	36



1.1.1	Identification des classes d'objets .....	36
1.1.2	Dictionnaire de donnée .....	37
1.1.3	Identification des attributs .....	38
1.1.4	Codification.....	41
1.1.5	Identification des associations.....	46
1.2	Construction du modèle dynamique .....	49
1.2.1	Scénarios et suivi d'événements.....	50
1.2.2	Les diagrammes d'états .....	57
1.3	Modèle fonctionnel .....	64
1.3.1	Diagramme à flot de données .....	65
2.	Conception de système .....	67
2.1	Règles de passage entre modèle objet et modèle tables relationnelles.....	67
2.2	Traduction du modèle objet en base de données relationnelles.....	68
3.	Conception des objets .....	74
	Conclusion.....	78

## **Chapitre V : Implémentation et Mise en œuvre**

	Introduction.....	79
1.	Environnement technique de développement.....	79
1.1	Présentation des langages de programmations utilisés.....	79
2.	Les différentes applications de logiciel.....	79
2.1	La phase d'authentification.....	80

### **Bibliographie**

### **Annexe**

## *Listes des Figures*

Figure I.1 les Niveaux du SGBD.....	3
Figure II.1 Base de données à Objets Multi-Niveaux.....	20
Figure II.2 BD Multi-Vues.....	23
Figure III.1 Scénario pour la gestion des transactions bancaires.....	30
Figure III.2 Diagramme de suivi d'événement pour la gestion des transactions Bancaires....	31
Figure III.3 Exemple de modèle dynamique.....	32
Figure III.4 Cycle de développement d'OMT.....	33
Figure IV.1 Diagramme de classe .....	48

*Introduction  
Générale*



## Introduction générale

Les bases de données sont actuellement au cœur du système d'information des entreprises, la notion de base de données est généralement couplée à celle de réseau afin de permettre à plusieurs utilisateurs d'accéder simultanément au même données via le réseau. Ces accès doivent être sécurisés contre la divulgation non autorisée de l'information, pour cela plusieurs recherches sont orientées vers le développement des modèles de sécurité mais la majorité de ces modèles ne traitent que le problème de confidentialité. Généralement les données n'ont pas tous les mêmes degrés de sensibilité, pour cela, des bases de données Multi-Niveaux dont l'objectif majeur est de gérer des données diversement classifiées pour des utilisateurs diversement habilités ont été développées. Le modèle multi-vues est un modèle de sécurité Multi-Niveaux pour gérer les base de données objet, la conception d'une base de données (Multi-Niveaux) en suivant ce modèle consiste à :

- Attribuer un niveau de classification a chaque données de la base selon son degré de sensibilité.
- Décomposer la base de données Multi-Niveaux en plusieurs fichiers physique de données Mono-Niveau, chaque fichier ne contient que des données ayant toutes le même niveau de classification

Notre but est de développer une base de données Médicale en appliquant le modèle Multi - Vues qui a été développé par A.Gabillon et F.Cuppens pour permettre la gestion et la sécurité (confidentialité) des données médicaux.

Pour ce fait nous allons utiliser :

- La méthode OMT pour la conception.
- Le SGBD Oracle 8.i pour l'implémentation de la base de données.
- Le logiciel C++ Builder pour la réalisation de logiciel.

Notre document se repartis comme suit :

- Introduction générale.
- Chapitre I : Généralité sur les bases de données et les SGBD.
- Chapitre II : La sécurité Multi- Niveaux et le modèle Multi- Vues.
- Chapitre III : Présentation de la Méthode OMT .
- Chapitre IV : Analyse et Conception.
- Chapitre V : Implémentation et Mise en œuvre.
- Conclusion

*CHAPITRE I*  
*Généralités sur*  
*les bases de données et*  
*les SGBD*



**Introduction:**

Les activités humaines génèrent des données. Il en a toujours été ainsi et, plus notre civilisation se développe, plus le volume de ces données croît. Aujourd'hui, les données sont de plus en plus souvent gérées par des moyens informatiques.

Les bases de données sont actuellement au cœur du système d'information des entreprises. Les systèmes de gestion de bases de données, peuvent maintenant être installés sur tous les types d'ordinateurs y compris les ordinateurs personnels.

Qu'est-ce donc qu'une base de données? Que peut-on attendre d'un système de gestion de bases de données? Et comment assurer la sécurité de celle-ci ? C'est à ces questions, entre autres, que ce chapitre essaie d'apporter des réponses.

**1. Définition d'une base de donnée:**

Une base de données (son abréviation est BD, en anglais DB, data base) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.

**Utilité des bases de donnée**

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers. Cela est d'autant plus utile que les données informatiques sont de plus en plus nombreuses. Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien répartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau

L'avantage majeur de l'utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément.



Le SGBD peut se décomposer en trois sous-systèmes:

- **le niveau interne (physique):**  
il permet le stockage des informations sur un support physique
- **le niveau conceptuel:**  
il gère l'ordonnancement des informations
- **le niveau externe:**  
il représente l'interface avec l'utilisateur

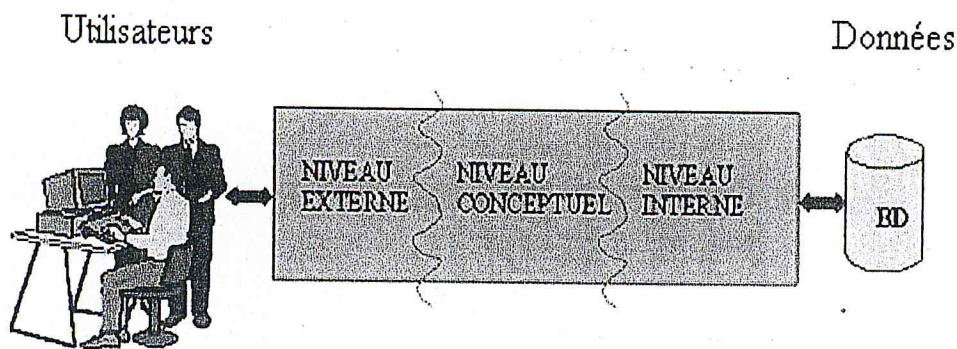


Figure I.1 les Niveaux du SGBD

## 2.1 les architectures d'un SGBD :

La plupart des SGBD suit l'architecture standard ANSI/SPARC qui permet d'isoler les différents niveaux d'abstraction nécessaires pour un SGBD.

D'un point de vue opérationnel, un SGBD est un ensemble de processus et de tâches qui supportent l'exécution du code du SGBD pour satisfaire les commandes des utilisateurs.

### 2.1.1 L'architecture client – serveur:

C'est une architecture hiérarchisée mettant en jeu d'une part un serveur de données gérant les données partagées en exécutant le code du SGBD avec d'éventuelles procédures applicatives. Autour de ce serveur s'articulent des processus attachés aux utilisateurs supportant les outils et les interfaces externes, les clients pouvant être organisés en différents niveaux supportant les applications et la présentation, et dans laquelle les clients dialoguent avec les serveurs via un

réseau en utilisant des requêtes de type question - réponse.

### 2.1.2 L'architecture repartis:

Afin de répondre à la tendance centralisatrice de l'approche client –serveur, certain SGBD préconise une architecture repartie.

Une architecture repartie fait interagir plusieurs serveurs gérant un ensemble de bases perçues comme une seule base par les utilisateurs

C'est une architecture composée de plusieurs serveurs coopérant à la gestion de base de donnée composée de plusieurs sous- bases gérées par un seul serveur, mais apparaissant comme des bases uniques centralisés pour l'utilisateur

## 2.2 Les caractéristiques du système de gestion de base de donnée:

D'une manière générale un SGBD doit avoir les caractéristiques suivantes:

- **Indépendance physique:** Le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que tous les aspects matériels de la base de données n'apparaissent pas pour l'utilisateur, il s'agit simplement d'une structure transparente de représentation des informations.
- **Indépendance logique:** le niveau conceptuel doit pouvoir être modifié sans remettre en cause le niveau physique, c'est-à-dire que l'administrateur de la base doit pouvoir la faire évoluer sans que cela gêne les utilisateurs .
- **Manipulabilité:** des personnes ne connaissant pas la base de données doivent être capables de décrire leur requêtes sans faire référence à des éléments techniques de la base de données.
- **Rapidité des accès:** le système doit pouvoir fournir les réponses aux requêtes le plus rapidement possible, cela implique des algorithmes de recherche rapides.
- **Administration centralisée:** le SGBD doit permettre à l'administrateur de pouvoir manipuler les données, insérer des éléments, vérifier l'intégrité des données.
- **Sécurité des données:** Le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs.
- **Limitation de la redondance:** le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes, afin d'éviter d'une part un gaspillage



d'espace mémoire mais aussi des erreurs .

- **Vérification de l'intégrité:** les données doivent être cohérentes entre elles, de plus lorsque des éléments font références à d'autres, ces derniers doivent être présents.
- **Partageabilité des données:** le SGBD doit permettre l'accès simultané à la base de données par plusieurs utilisateurs.
- **gérer les droits d'accès aux données selon les utilisateurs.**

### 3. Le modèle relationnel de données :

Le modèle relationnel a été formalisé par CODD. Dans ce modèle, les données sont stockées dans des tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Un ensemble de données sera donc modélisé par un ensemble de tables.

De façon informelle, on peut définir le modèle relationnel de la manière suivante :

Les données sont organisées sous forme de tables à deux dimensions, encore appelées relations et chaque ligne n-uplet ou tuple, Les données sont manipulées par des opérateurs de l'algèbre relationnelle, L'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.

Au modèle relationnel est associée la théorie de la normalisation des relations qui permet de se débarrasser des incohérences au moment de la conception d'une base de données.

Le succès du modèle relationnel auprès des chercheurs, concepteurs et utilisateurs est dû à la puissance et à la simplicité de ses concepts. En outre, contrairement à certains autres modèles, il repose sur des bases théoriques solides, notamment la théorie des ensembles et la logique mathématique (théorie des prédicats d'ordre 1).

#### 3.1 Les objectifs du modèle relationnel :

- Proposer des schémas de données faciles à utiliser
- Améliorer l'indépendance logique et physique
- Mettre à la disposition des utilisateurs des langages de haut niveau pouvant éventuellement être utilisés par des non informaticiens
- Optimiser les accès à la base de données
- Améliorer l'intégrité et la confidentialité
- Fournir une approche méthodologique dans la construction des schémas.

#### 3.2 Base de données relationnelles :

Une base de données relationnel est une Base de données dont le schéma est un



ensemble de schémas de relations et dont les occurrences sont les tuples de ces relations. D'autre part Une base de données relationnelle est une base dans laquelle on gère les liaisons logiques entre plusieurs structures de données. Cela permet de simplifier le développement et supprimer les informations redondantes du système d'informations notamment pour que ce dernier soit facilement mis à jour.

#### **4. Les extensions du modèle relationnel et les modèles objets :**

Les SGBD relationnels répondent mal à certaines applications gérant des objets complexes (textes, graphiques, cartes, images, données multi -dimensionnelles) et des objets dynamiques.

Un autre problème important soulevé par le relationnel est celui du langage de requêtes qui délivre des ensembles de données (tuples) aux programmes d'application qui eux doivent relire et traiter ces tuples un par un.

Ces limites du modèle relationnel ont conduit à de nombreuses recherches sur de nouveaux modèles de bases de données et, actuellement, deux types d'alternatives sont proposés : le modèle relationnels étendus et les modèles orientés objets.

Dans les deux cas, l'enrichissement du modèle relationnel est fait par l'approche "objet", tirée des concepts des nouveaux langages de programmation dits "orientés objet". Le modèle relationnel étendu va ajouter des fonctionnalités de type objets aux SGBD relationnels classiques.

Le modèle orienté objet va conduire à de nouveaux SGBD, dits SGBD orientés objets ou SGBD-OO, s'appuyant sur la programmation objet en lui intégrant des fonctionnalités de SGBD.

Dans les modèles relationnels étendus, on trouve de nouvelles possibilités de définition et de manipulation des constituants des tables : on peut créer ses propres types de données, appelés ADT (types de données abstraits ou Abstract Data Type).

Ces types d'objets définissent à la fois les domaines de valeurs des objets et les opérations que l'on peut leur appliquer, c'est le concept d' "encapsulation" des objets. A ces nouvelles fonctionnalités il faut adjoindre des extensions du langage de requêtes.

L'autre approche, celle des SGBD-OO, consiste à proposer un nouveau modèle de base "orienté objet" qui puisse être géré par un système de type SGBD.

Il n'existe pas, contrairement au modèle relationnel, de fondement formel et théorique

permettant de donner une définition universelle du modèle objet et chaque système offre une solution spécifique.

Toutefois, il existe des concepts communs que l'on doit retrouver dans tout modèle orienté objet.

Les SGBD-OO apportent sans aucun doute de nombreux avantages par rapport aux SGBD relationnels classiques : facilités de manipulation d'objets complexes, extensibilité des types de données, même formalisme de stockage des données et programmes, hiérarchie des classes d'objets et mécanisme d'héritage.

En revanche, ils n'ont plus la simplicité du modèle relationnel, le principe de la manipulation

non procédurale des données, et l'avantage de la standardisation.

La migration vers un SGBD-OO ne s'impose que pour des applications particulières où les performances du relationnel, aussi bien en temps machine qu'en temps de développement seraient inacceptables.

## **5. Modèle orientée objet :**

Les modèles à objets, encore appelés modèle orienté objet ou simplement modèle objet sont issus des réseaux sémantiques et des langages de programmation orienté objet. Ils regroupent les concepts essentiels pour modéliser de manière progressive des objets complexes encapsulés par des opérations de manipulation associées, il vise à permettre la réutilisation de structure et d'opération pour construire des entités plus complexes.

### **5.1 L'approche objet :**

Prenant une démarche inverse à la décomposition en relation disjointe, l'approche objet tente de représenter de façon naturelle des objets dans toute leur complexité, aussi bien descriptive que procédurale. Un objet permet de regrouper sous un même élément une structure de description (des attributs) et un comportement (sa façon d'agir ou de réagir, décrit par des méthodes).

Les attributs peuvent être élémentaires, c'est-à-dire d'un type simple (entier, réel, date, etc.), mais peuvent également être d'autres objets. L'un des intérêts majeurs de l'approche objet consiste dans sa capacité à décrire et gérer les entités du monde réel en un formalisme hiérarchique (des attributs sont eux-mêmes des objets), et en associant



données et procédures. La notion de classe correspond au schéma d'un objet (description des attributs et des méthodes).

### 5.2 Base de données orientées objet:

Une base de données orientée objet permet de stocker des données, et leurs comportements, c'est -à- dire la façon dont elles interagissent.

### 5.3 SGBD orienté-objet :

Les bases de données orientées objet existent depuis le milieu des années 1980. Un SGBDOO est construit pour gérer des collections d'objets, suivant un schéma de données correspondant à la définition de classe d'objets.

Leur objectif principal est d'unifier deux technologies : les systèmes de gestion de bases de données et la programmation orientée objet. Un langage de programmation orienté objet est en général très expressif mais manque d'outils performants de gestion de la persistance.

### 5.4 Objectifs des SGBDOO :

- Réduire le dysfonctionnement entre le langage de programmation et langage de base de données en offrant un unique langage de programmation de base de données.
- Supporter directement les objets arbitrairement complexes par un modèle objet.
- Partager le code réutilisable au sein du SGBD.

## 6. Problématique de la sécurité:

Généralement, on caractérise la sécurité des bases de données par trois propriétés [Com90] :

- **La confidentialité** : prévention contre des divulgation non autorisées de l'information.
- **L'intégrité** : Prévention d'une modification non autorisée de l'information.
- **La disponibilité** : prévention d'un déni non autorisé d'accès à l'information ou a des ressources.

Pour choisir un système de base de données sécurisé, l'utilisateur a besoin d'éléments de comparaison lui permettant de choisir un SGBD parmi ceux qui existent.

Ce qui a donné naissance un nouveau domaine de recherche dont le but est de modéliser formellement le concept de sécurité.

Les objectifs d'un modèle de sécurité sont les suivants :

- Exprimer les besoins de sécurités dans un contexte informatique.
- Fournir des moyens permettant de convaincre que les besoins sont satisfaits.



- Fournir des méthodes permettant de concevoir et d'implanter le système.

Plusieurs modèles de sécurité ont été développés, la majorité ne traite que le problème de confidentialité des informations (par exemple [Rei75]). Les modèles qui traitent les autres problèmes sont moins nombreux.

Les trois propriétés de sécurité, c'est-à-dire confidentialité, intégrité et disponibilité partagent un point commun qui est la notion de règlement de sécurité.

Toute organisation utilise des règlements de sécurité pour définir comment les membres de l'organisation peuvent accéder aux informations de cette organisation.

Le livre « orange » [Dei83] a proposé une classification des règlements de sécurité en deux grandes catégories :

Le contrôle d'accès discrétionnaire et le contrôle d'accès obligatoire.

Dans ces deux approches, l'unité d'information (ou le container d'information) qui a besoin d'être protégée peut varier de différentes façons, elle peut être une base de données entière, des ensemble de relation ou bien un attribut particulier d'un n-uplet particulier dans une relation particulière.

### 6.1 Contrôles d'accès discrétionnaires :

Dans les contrôles d'accès discrétionnaires, un utilisateur donné aura typiquement différents droits d'accès (appelé aussi privilèges ou autorisations) sur différents objets. De plus des utilisateurs différents pourront typiquement avoir des droits différents sur le même objet. C'est une approche flexible.

Dans ce type de contrôle, l'utilisateur possède d'une part des droits d'accès à l'information, et d'autre part, des droits de transmission de droits. Ce qui veut dire, que les utilisateurs sont autorisés à définir leurs propre règlement de sécurité en attribuant et en retirant des droits d'accès au autre membres de l'organisation.

### 6.2 Contrôles d'accès obligatoires :

Dans cette approche, on considère que les informations appartiennent à un organisme, ces informations sont stockées dans divers containers d'informations et ces containers sont classifiés avec différents degrés de sensibilité.

D'après leurs rôles à l'intérieurs de l'organisme, les utilisateurs reçoivent une habilitation qui leurs permet d'accéder à certains containers et de connaître les informations.

A partir de ça, on peut définir informellement la confidentialité de la façon suivante : un utilisateur ne doit pas connaître d'information pour lesquelles il n'a pas reçus d'habilitation, et un container d'information donné ne peut alors être accédé que par des utilisateurs ayant le même niveau d'habilitation approprié c'est une approche plus rigide. Les niveaux de classification et d'habilitation sont supposés former un ordre strict (par exemple très secret>secret>confidentiel>public, etc.).

Les règles à respecter sont les suivantes :

1-un utilisateur *i* peut observer un container *j* seulement si le niveau d'habilitation de *i* est supérieur ou égal au niveau de classification de *j*

2-Un utilisateur *i* peut modifier un container *j* seulement si le niveau d'habilitation de *i* est égal au niveau de classification de *j*.

Contrairement au contrôle d'accès discrétionnaire, dans le contrôle d'accès obligatoire, seul l'organisation est autorisée à donner ou à enlever un droit d'accès à l'un de ses membres.

Un règlement multi-niveaux est un exemple de gestion d'accès obligatoire.

Les individus reçoivent un niveau d'habilitation et chaque container d'information reçoit un niveau de classification. Le niveau résume la confidentialité.

L'utilisateur est autorisé à connaître une information si le niveau d'habilitation de l'information domine le niveau de classification du container dans le quel est rangée l'information.

**Conclusion :**

Dans ce chapitre nous avons présenté des généralités sur les bases de données et les SGBD et leurs caractéristiques.

En second lieu nous avons introduits les concepts essentiels qui supportent le modèle relationnel, car il fait aujourd'hui autorité dans l'industrie issue de la théorie des relations, il est à l'origine une remarquable construction de la recherche, il a su progressivement intégrer des concepts de plus en plus riches, tel que l'intégrité, référentielle, les règles actives etc.

Le modèle a aujourd'hui intégré les concepts de l'objet pour fonder l'objet relationnel, sans oublier de mentionner que les SGBD OO, offrent pour la plupart des environnements de développement visuels élaborés.

La sécurité des bases de données semble être un problème important sur le quel des efforts de recherches considérables devront être consacrés dans le futur ; les informations dans un système sont souvent très sensibles, il est donc important de pouvoir détecter les risques de divulgations et de développer des méthodologies pour réduire ces risques.

D'où l'apparition des systèmes d'informations à base de règlement multi-niveaux, et quand va traiter dans le chapitre suivant.



*CHAPITRE II*  
*la sécurité Multi-Niveaux*  
*et le modèle Multi-Vues*

## **Introduction**

La sécurité multi-niveaux a vu le jour dans les organisations militaire bien avant l'apparition de l'informatique dans leurs organismes.

L'objectif d'une base de donnée multi-niveaux est de gérer des données diversement classifiées et d'assurer la confidentialité de ces données vis-vis d'utilisateurs diversement habilité. [CUP00]

Dès le début, plusieurs modélisation de sécurité en vus le jour (voir par exemple [Gro76]), tel que le modèle de Bell et Lapadula [Bel75]. Plusieurs types d'architecture pour une base de donnée multi-niveaux on été envisagée [Boa83]. L'approche filtre, l'approche noyau de sécurité, l'approche sujet de confiance et l'approche répliquée.

### **1. les approches de sécurité Multi-niveau :**

#### **1.1 L'approche filtre :**

Est la solution la plus simple qu'on puisse envisager pour sécuriser une base de donnée. Il s'agit d'ajouter un filtre à un SGBD existant. Le premier rôle du filtre consiste à authentifier les utilisateurs pour connaître le niveau d'habilitation. Son second rôle est d'intercepter toutes les requêtes émises par les utilisateurs et toutes les réponses fournis par le SGBD.

Donc, le filtre a deux fonctionnalités principales :

- Modifier les requêtes des utilisateurs pour que l'évaluation de la requête ne fournisse comme résultats que des données ayant un niveau de classification inférieur ou égale au niveau d'habilitation de l'utilisateur.
- Filtrer le résultat de manière à éliminer toutes les données qui n'auraient pas un niveau de classification inférieur ou égale de l'utilisateur.

#### **1.2 L'approche noyau de sécurité :**

Consiste à développer un SGBD en s'appuyant sur un système d'exploitation multi-niveaux existant pour réaliser les contrôles sécurité. En particulier, les fonctions d'authentification et de contrôle d'accès sont réalisées par le noyau de sécurité du système d'exploitation multi-niveaux.

Comme les systèmes d'exploitation multi-niveaux disponible ne permettent de gérer que des fichiers mono-niveau (c'est-à-dire, chaque fichier ne contient que des données ayant toutes les

même niveau de classification), il faut partitionner les données du SGBD pour qu'elles puissent être gérées dans les fichiers mono-niveau.

### 1.3 L'approche sujet de confiance :

Est une variante de l'approche noyau de sécurité dans laquelle la plupart des contrôles de sécurité ne sont pas réalisés par le système d'exploitation multi-niveau sous-jacent mais par le SGBD. Le SGBD inclut un noyau de sécurité qui effectue l'authentification des utilisateurs et le contrôle d'accès aux données de l'application. Dans cette approche, les données du SGBD sont regroupées dans un fichier unique.

### 1.4 L'approche répliquée :

Consiste à utiliser un SGBD séparé pour gérer toutes les données de niveau de classification inférieurs à un niveau de sécurité donné. Le serveur du SGBD est donc répliqué, il y a autant de serveurs de SGBD qu'il y a de niveau de sécurité, chaque serveur étant associé à un niveau de sécurité. Les données sont également répliquées, une donnée de niveau N est répliquée vers tous les serveurs de niveaux supérieurs à N. Pour développer un SGBD suivant cette approche il faut implémenter un frontal, c'est-à-dire, un programme de confiance qui réalise les contrôles de sécurité. Il authentifie les utilisateurs et transmet leurs requêtes vers le serveur correspondant au niveau d'habilitation de l'utilisateur. Le frontal doit aussi contrôler la répliquée des données.

Au vu des avantages et des inconvénients de chaque approche, l'architecture idéale en matière de SGBD multi-niveaux n'existe pas, étant donné que la sécurité est une affaire de compromis entre de nombreux paramètres [Cup97].

## 2. La confidentialité :

La propriété de la confidentialité en sécurité multi-niveau s'exprime de la façon suivante :

« Un individu ne peut connaître l'information contenue dans un objet que si l'habilitation de cet individu est supérieure ou égale au niveau de classification de l'objet » [CUP00].

Dans un système informatique, les deux propriétés suivantes sont nécessaires mais pas suffisantes pour assurer la propriété de confidentialité :

• **No Read Up** : Un sujet ne peut avoir un droit de lecture que si son niveau de sécurité est supérieur ou égal au niveau de classification de l'objet.



● **No Write Down:** Un sujet ne peut avoir un droit d'écriture sur un objet que si son niveau de sécurité est inférieur ou égale au niveau de classification de l'objet.

### 3. Granularité de la classification :

Un premier problème à considérer lorsque l'on développe une application multi-niveaux et celui de la granularité de la classification, c'est-à-dire, le grain d'information qui recevra une classification.

Dans le cas d'une base de données relationnelles, plusieurs granularités ont été proposées dans la littérature : la relation, le n-uplet, et l'attribut de n-uplet.

● L'interprétation de la granularité « n-uplet » est simple. Si l'on attribue un niveau de classification  $n$  à un n-uplet cela signifie que l'information représentée par le n-uplet est elle-même classée au niveau  $n$ .

Par exemple :

- 1- la relation employée (nom, Age, salaire). Si on attribue le niveau de classification secret au n-uplet (Jean, 30, 1500), cela signifie que l'information « Jean est un employé ayant 30 ans et gagnant 1500 £ » est classée au niveau secret.
- 2- la relation diagnostic (Numéro diagnostic, Nom Maladie, Date diagnostic, Nom médecin soignant, Service, Analyses et radio, Description, Risque de Contamination, Mode d'entrée). Si on attribue le niveau de classification secret au n-uplet (C200200001001, Cancer, 12/10/2003, Benstali Karima, Anti-cancer, Analyses et Radio, Description, Nom, hospitalisation), cela signifie que l'information « Le patient ayant le Diagnostic identifié par le matricule C200200001001 décrivant la maladie "Cancer" et ne cause pas de risque de contamination ayant fait les radio YY, hospitalisé le 12 octobre 2003 dans le service de cardiologie soigné par le médecin Benstali Karima » est classée au niveau secret.

● L'interprétation de la granularité « relation » est moins évidente. L'interprétation la plus courante est de considérer que si l'on attribue un niveau de classification  $n$  à une relation  $R$ , cela signifie que tous les n-uplets de la relation sont eux-mêmes classés au niveau de  $n$ . C'est l'interprétation qui convient lorsque les n-uplets ne reçoivent pas un niveau de classification. Dans le cas contraire, une autre interprétation est de considérer que le niveau  $n$  représente la classification de l'information «  $R$  est une relation de la base de données ».

Cette deuxième interprétation est celle utilisée par le modèle Seaview [DLS87].

- La granularité « attribut de n-uplet » est celle retenue dans de nombreux projets de base de données multi-niveaux relationnels, par exemple, si P, S ce sont deux niveaux de sécurité correspondant respectivement à public, secret alors on peut considérer les n-uplet :

- Employé (jean, P, 1500, S).

- Diagnostic (C200200001001, P, Cancer, S, '12/10/2003', P, Benstali Karima, P, Anti-cancer, P, Analyses et radio, S, Description, S, Nom, P, hospitalisation, P).

Dans [CuY92], on peut trouver l'interprétation suivante de la granularité « attribut de n-uplet » : si l'on affecte un niveau de classification  $N_i$  à l'attribut  $A_i$  d'un n-uplet, alors  $N_i$  représente la classification de l'information correspondant à l'association entre les attributs  $A_k$  et  $A_i$ , ou  $A_k$  représente l'ensemble des attributs constituant la clé de la relation dans le cas où  $A_i$  est un attribut appartenant à la clé de la relation, alors  $N_i$  représente la classification de l'information contenue dans la clé de la relation.

Par exemple, si l'on considère que l'attribut nom est la clé de la relation employé, alors on peut donner l'interprétation suivante pour le n-uplet.

Exemple1 :

Employé (jean, P, 1500, S).

1. L'information «jean est un employé » est classée au niveau P.
2. L'information « jean gagne 15000€ » est classée au niveau S.

Et si l'on considère que l'attribut Matricule est la clé de la relation Diagnostic, alors on peut donner l'interprétation suivante pour le n-uplet.

Exemple2 :

Diagnostic(C200200001001,P,Cancer,S,12/10/2003,P,BenstaliKarima,P,Anti-cancer,P, Analyses et radios,S,Description ,S, Nom ,P, hospitalisation,P).

1. L'information «Le patient ayant le Diagnostic identifié par le matricule 'C200200001001' et n'entraîne pas de risque de contamination hospitalisé le '12 octobre2003' dans le service Anti-cancer écrit par le médecin Benstali Karima »est classée au niveau P.

2. L'informations Le patient ayant le Diagnostic identifié par le matricule 'C200200001001' décrivant la maladie' Cancer' avec la Description 'XXX' ayant fait les radios 'YY'est classé secret.



En s'appuyant sur cette interprétation on peut étudier le problème de la décomposition d'une base de données multi-niveaux en un ensemble de base de données mono-niveaux. Le problème de la granularité se pose également pour les bases de données orientées objet.

#### 4. Gestion des leurres :

Intuitivement, un leurre est une information fausse introduite dans une base de donnée multi-niveau, en général, pour protéger l'existence d'une information sensible.

Par exemple, supposant que la base de données contienne l'information

Diagnostic(C200200001001,Cancer) c'est-à-dire « Le patient qui a le Diagnostic identifié par le matricule 'C200200001001' décrivant la maladie Cancer ».

Supposant que cette information soit classé au niveau secret :

- [secret] Diagnostic (C200200001001, Cancer)

Ou [secret] X signifie que l'information représentée par la formule X est classée au niveau secret.

Supposant également que la maladie décrite dans le diagnostic soit unique et que la base de donnée contient l'information Diagnostic(C200200001001,Anémie) et que cette information soit classée au niveau public :

- [public] Diagnostic (C200200001001, Anémie)

Comme un diagnostic donné ne peut pas décrire deux maladies à la fois, cette dernière information est en générale interprétée comme un leurre. Les leurres sont nécessaires dans une base de données multi-niveaux dans des situations bien particulières.

Dans notre exemple, supposant que le patient X voudrai savoir sa maladie et son niveau d'habilitation est public. La base de données ne peut naturellement pas répondre « la maladie décrite dans le diagnostic correspondant au patient X est un cancer » puisque cette information est secrète. Supposant que la base de données répondent « vous n'avez pas le droit de connaître cette information ».

Le patient X peut alors déduire de cette réponse que sa maladie est classée secret.

Dans ce cas, la base de données ne peut pas répondre au patient X « vous n'avez pas le droit de connaître cette information ». L'utilisation d'un leurre est alors nécessaire et la base de donnée multi-niveaux va alors répondre, par exemple, que la maladie de patient X est Anémie.



Dans la littérature, la gestion du leurre est souvent confondue avec la technique de poly instanciation. Cette technique qui a été introduite la première fois dans le projet Seaview [DLS87], consiste à accepter plusieurs n-uplets différents mais ayant la même valeur des attribut constituant la clé, pourvu que les n-uplets aient des classifications différentes.

Dans ce cas, le n-uplet ayant la classification la plus basse est implicitement interprété comme un leurre. Malheureusement, cette approche ne convient pas pour gérer des leures lorsque les niveaux de sécurité sont particulièrement ordonnés. Par exemple, supposant que la base de données contienne les deux faits suivant :

- Diagnostic (C200200001001, Cancer) classé au niveau C1.
- Diagnostic (C200200001001, Anémie) classé au niveau C2.

Si l'on suppose que la maladie décrite dans un diagnostic donné est unique, l'un de ces deux faits est faux mais si les niveaux C1 et C2 sont incomparables, on ne peut pas implicitement déduire lequel de ces deux faits est un leurre.

### **5. Inférence non autorisée d'information :**

Dans cette section, nous supposons pour simplifier la présentation que le règlement de sécurité multi-niveaux n'utilise que deux niveaux pour classer les informations : un niveau public note P, un niveau secret note S.

Le problème de l'inférence non autorisée de l'information peut se définir informellement de la façon suivante : Est-il possible de déduire des informations classées secrètes en utilisant les informations classées publiques, Toutefois, il convient d'apporter plusieurs précisions.

- Un utilisateur habilité public peut avoir accès à des informations qui sont externes à la base de données, qui une fois combinées avec les informations gérées par la base de données publique, permettent à cet utilisateur de déduire des informations secrètes. Pour résoudre ce problème, on suppose que la base de données publique gère également « un modèle de l'utilisateur public ». Ce modèle devrait contenir toutes les règles et les connaissances communes que les individus publics possèdent sur le modèle représenté dans la base. Une telle hypothèse étant en générale irréaliste, le modèle de l'utilisateur devra approcher « au mieux » cette représentation idéale.

- la deuxième précision qu'il est important de faire concerne la définition même du concept d'information secrète. Pour analyser correctement le problème d'inférence non autorisée d'information dans une base de données multi-niveaux, il est essentiel de pouvoir

donner une caractérisation précise des informations qui doivent être considérées secrètes dans la base de données. Il est possible d'être plus au moins exigeant sur l'ensemble d'informations que l'on veut protéger, suivant que l'on interdit seulement la divulgation exacte des informations secrètes ou suivant que l'on interdit également la divulgation disjonctive, négative ou existentielle des informations secrètes.

Un utilisateur peut construire divers type de raisonnement pour arriver à déduire une information secrète. Ceci augmente beaucoup la complexité de traitement du problème de l'inférence. On présente trois classes de canaux [Gal92] :

- Canal déductif** : Il s'agit d'un canal qui permet à un utilisateur public d'inférer des informations interdites en ne mettant en œuvre qu'un raisonnement classique.
- Canal abductif** : Un canal abductif existe lors qu'un raisonnement classique utilisant les informations explicitement stockées dans la base de données ne permet d'inférer d'informations interdites, mais ce raisonnement pourrait être complété en faisant des hypothèses sur certaines informations.
- Canal probabiliste** : Ce type de canal existe lorsqu'un utilisateur parvient à estimer la vraisemblance d'une information secrète en utilisant les informations publiques. Cette liste n'est pas exhaustive et l'on peut imaginer d'autres types de canaux fondés sur d'autres types de raisonnement.

## 6. Représentation du Modèle Multi-Vues :

Après avoir défini les notions principales de la sécurité multi-niveaux. Plusieurs modèles de sécurité multi-niveaux ont vu le jour depuis leurs apparition, nous allons présente un modèle de sécurité multi-niveaux qui est modèle multi-vues de F.Cuppens et A.Gabillon.

### 6.1 Définition

Le modèle Multi-vues est un modèle de sécurité multi-niveaux pour les bases de données objets, développé par A.Gabillon et F. Cuppens [CGY93 a].

Pour concevoir ce modèle, les auteurs ont du répondre à deux questions.

- A quel type d'informations dans une base de données à objets, peut on associer un niveau de classification ? certaines approches considèrent que seul un objet peut recevoir un niveau de classification partageant toute l'information encapsulée dans l'objet (approche dite mono-niveau) [Jak 90], alors que d'autres approches considèrent qu'un objet est une entité



complexe et que ses attributs peuvent être classifiés distinctement (Approche dite multi-niveaux) [KTT89].

▪ Comment décomposer la base de données multi-niveaux en une collection de fichiers physiques mono-niveaux (étant donné que tout noyau de sécurité de système opératoire existant ne sait gérer que des fichiers mono-niveaux) ?

Comme réponse à la première question, Gabillon et Cuppens ont utilisé pour leur modèle l'approche « objet multi-niveaux » qui permet d'assigner un niveau de classification à tout type d'information qu'il pouvait être pertinent de protéger dans une BDO [CuG98][CuG99]. Donc le modèle de sécurité autorise la protection des éléments suivants :

- Protection des objets :
  - Classification de l'existence des objets.
  - Classification des valeurs d'attributs.
  - Classification de l'existence des liens d'instanciation.
- Protection du schéma :
  - Classification de l'existence des classes.
  - Classification de l'existence des attributs.
  - Classification de l'existence des méthodes.
  - Classification de l'existence des liens d'héritage.

Pour résoudre le second problème, les auteurs ont décomposé une base de données conceptuelle multi-niveaux en autant de base de données mono-niveau que de niveaux de sécurité.

### 6.2 Exemple :

Nous supposons que les attributs sont associés à des valeurs atomiques, et que nous n'avons que deux niveaux de sécurité Public et Secret. Les informations secrètes sont en gras.



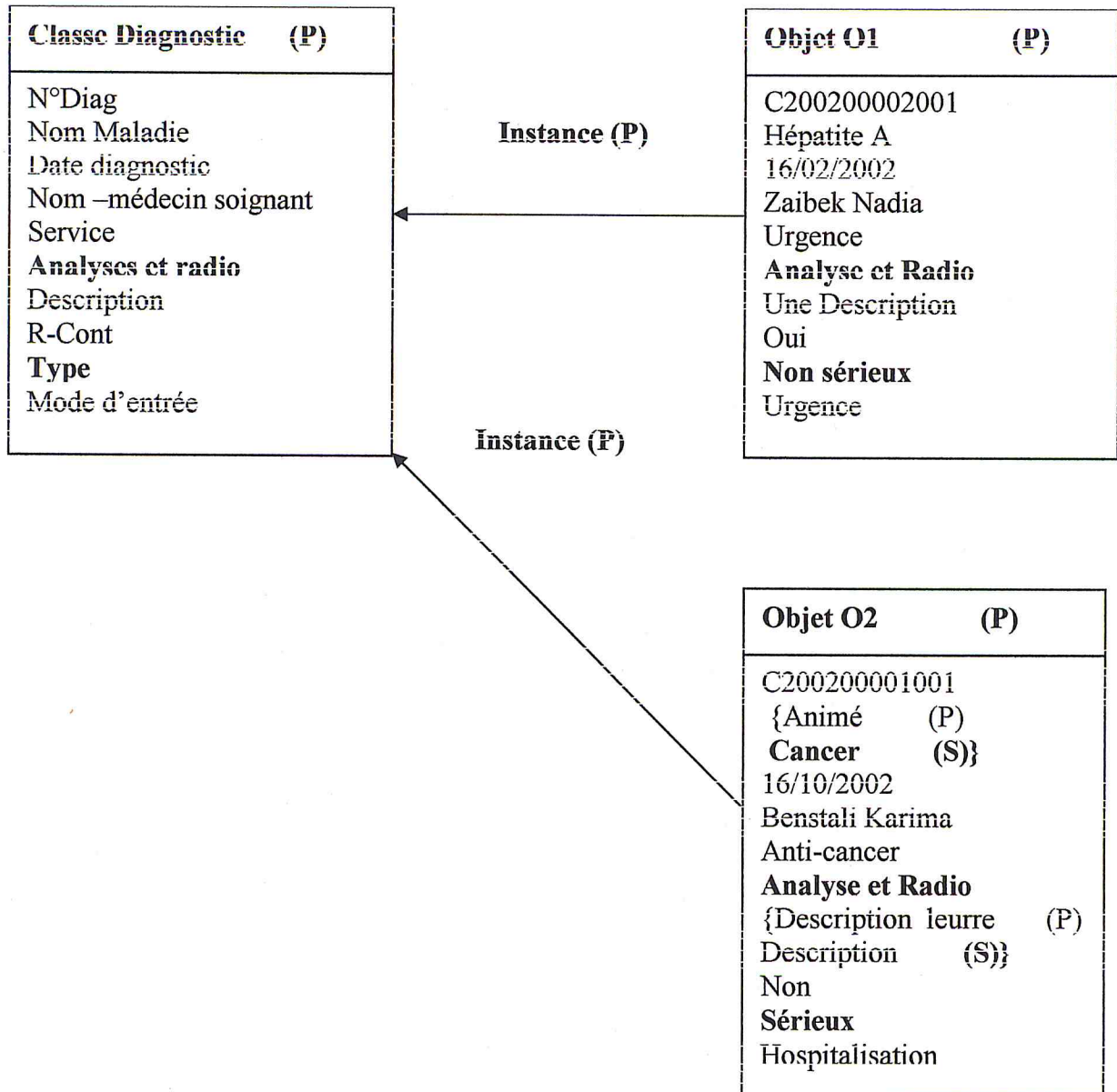


Figure II.1 Base de données à Objets Multi-Niveaux

Cet exemple comprend une classe et deux objets. L’existence de la classe Diagnostic est publique l’existence des attributs Numéro diagnostic, Nom Maladie, Date diagnostic, Nom –médecin soignant, Service, Description, Risque de Contamination, Mode d’entrée est publique.

Par contre l'existence des attributs Type, Analyse et Radio est secrète. Seuls les utilisateurs habilités Secret auront la possibilité de connaître l'existence de ces attributs.

L'existence des deux objets O1 et O2 n'est pas protégée. Numéro diagnostic, Nom Maladie, Date diagnostic, Nom –médecin soignant, Service, Description, Risque de Contamination, Mode d'entrée sont des informations publiques. Les valeurs que peut prendre les deux attributs secrets Type, Analyses et radio sont aussi des informations secrètes.

Les auteurs ont établi une liste de théorèmes dits de *Contrôle de flux* qui doivent être respectés lors de l'affectation des niveaux de sécurité. L'un de ces théorèmes stipule que le niveau de sécurité associé à une valeur d'attribut doit être supérieur ou égal au niveau de sécurité protégeant l'existence de l'attribut.

Etant donné, que l'attribut Type est classifié Secret donc la valeur qui lui associée sérieux ou non sérieux sont classifié aussi secrets, car rappelons le, dans notre exemple il n'existe que deux niveaux de classification Secret et Public.

L'attribut maladie est un attribut particulier. Cet attribut est appelé un *attribut polyinstancié* c'est-à-dire un attribut associé avec plusieurs valeurs chacune classifiée différemment. La vraie maladie de l'objet O2 est Cancer.

Cette information est considérée comme étant très sensible, pour cela, il est exclu qu'un utilisateur habilité public voulant accéder à la valeur de l'attribut maladie se voit refuser l'accès en lecture, de même pour l'attribut Description.

Ce refus éveillerait les soupçons de cet utilisateur habilité public qui pourrait alors deviner l'existence d'une information secrète, une seconde valeur classifiée publique est introduite. Cette seconde valeur, qui est appelée un *leurre*, est fournie à tout utilisateur habilité public tentant de connaître la maladie de l'objet O2.

O1 et O2 sont des instances de la classe Diagnostic et ce sont des informations non protégées. O1 est un Diagnostic non sérieux et c'est aussi une information publique. Par contre, O2 est un Diagnostic Sérieux est une information secrète.

La base de donnée de la figure II.1 est purement logique, l'implanter dans un fichier multi-niveaux est irréaliste, pour cela, Gabillon et Cuppens ont tenté de décomposer la base de données conceptuelle multi-niveaux en autant de bases de données mono-niveau que de niveaux de sécurité. Le principe est le suivant :

-Toute information de niveau  $l$  est répliquée dans toutes les bases de données mono-niveau de niveau supérieur ou égal à  $l$ . Ce principe souffre d'une exception : Une valeur d'attribut de niveau  $l$ , mais n'est pas répliquée au niveau immédiatement supérieur s'il existe une autre valeur pour le même attribut classifié au niveau immédiatement supérieur (cas d'un attribut polyinstancié).

-Le concept d'identificateur est étendu. Tout objet  $o$  (respectivement classe  $c$ ) inséré dans une base de données de niveau  $l$  est identifié par le couple  $(o, l)$  (respectivement  $(c, l)$ ).

-Un lien d'héritage est introduit entre toute classe  $c$  au niveau  $l$  et cette même classe  $c$  au niveau immédiatement supérieur. Ce principe permet de répliquer automatiquement attributs et méthodes.

-Si un attribut d'objet est inséré dans la base de données de niveau  $l$  n'a pas de valeur classifiée au niveau  $l$  alors la valeur de cet attribut devient un pointeur vers la valeur insérée dans la base de données de niveau immédiatement inférieur à  $l$ .

L'application de ces principes à la base de données de la figure II.1 permet d'obtenir la base de données multi-vues suivante :



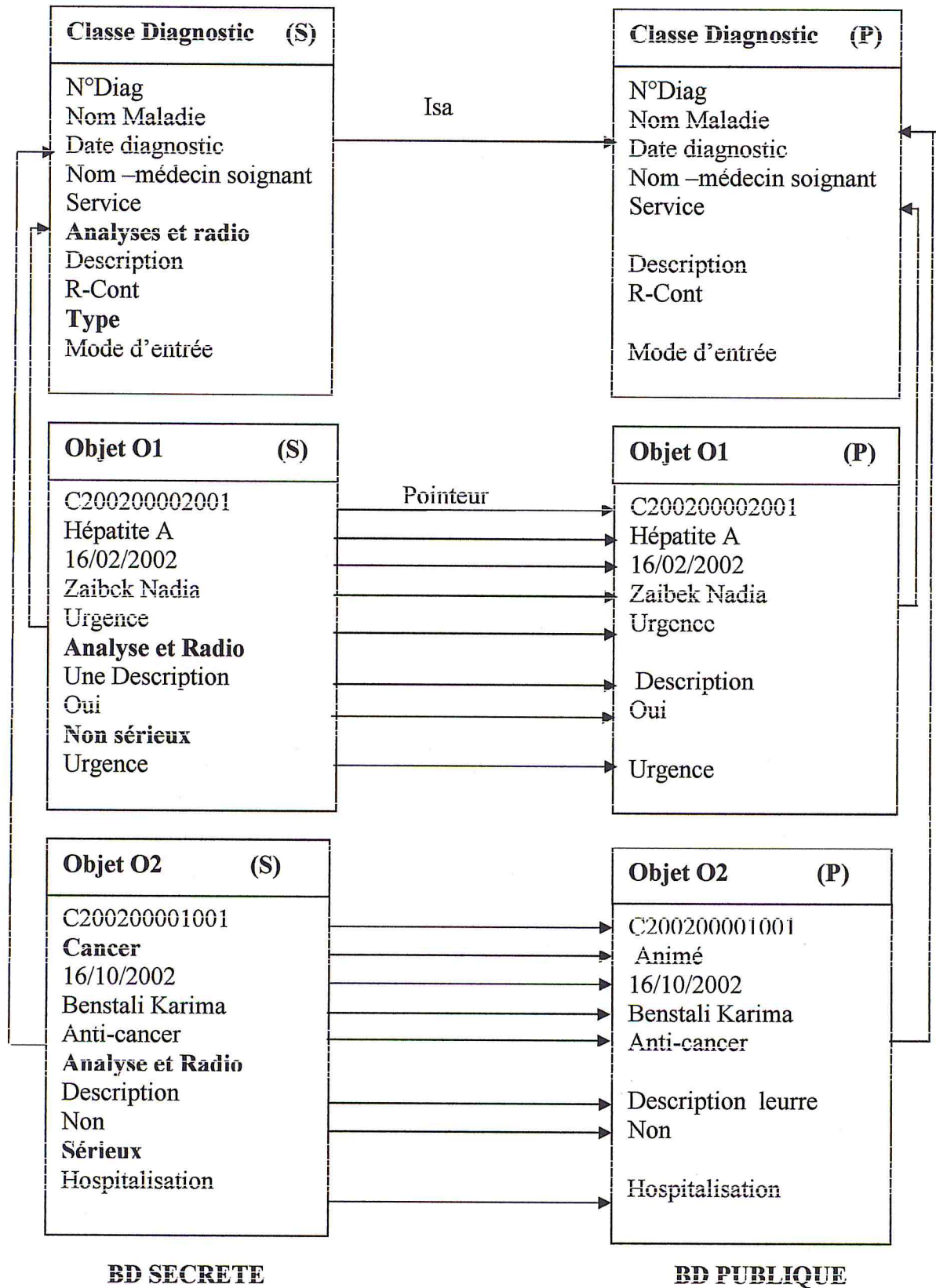


Figure II.2 BD Multi-Vues

Sur l'exemple de la figure II.1, on a supposé n'avoir que deux niveaux de sécurité. Donc sur cet exemple la base de données Multi-vues comprend deux bases de données mono-niveau.

Chaque information classifiée au niveau Secret sur la figure II.1 est insérée dans la base de données secrète. Chaque information classifiée au niveau Public sur la figure II.1 est insérée dans la base de données publique et est répliquée dans la base de données secrète.

La valeur **Anémie** de l'attribut Maladie de l'objet O2 n'est pas répliquée au niveau secret car rappelons le, cette valeur est un leurre qui est contredit par la valeur **Cancer** au niveau secret.

La base de données Multi-Vues comprend maintenant 2 classes ((Diagnostic, S), (Diagnostic, P)) Et 4 objets ((O1, P), (O1, S), (O2, P), (O2, S)).

La réplication des valeurs d'attributs (non leurre) du niveau public au niveau secret est automatiquement assurée par des pointeurs associés aux attributs correspondants dans la base de données de niveau Secret.

## Conclusion

D'après F.Cuppens et A.Gabillon, le modèle Multi-Vues est un modèle particulièrement complet avec un grand pouvoir d'expression qui surpasse les autres réalisations étant donné qu'il permet de considérer qu'un objet est une entité complexe et dont ces composants pouvait avoir différents niveaux de classifications [Jak90][KTT89].

**Introduction :**

Beaucoup de travaux existent aujourd'hui sur l'approche orientée objet, qui tient une place importante dans les langages de programmations et les systèmes d'exploitations que celui des systèmes d'informations et des bases de données.

Les récents développements méthodologies, « méthodes orientées objets », ont favorisé l'émergence de modèles permettant de mieux cerner la complexité des systèmes d'informations en spécifiant l'aspect statique et l'aspect dynamique.

Il existe actuellement plusieurs méthodes orientées objets, dont certaines sont dédiées à l'analyse, d'autres à la conception et il y a celles qui considèrent les deux aspects. On citera parmi elles : **OOD, HOOD, OOA, OOA/OOD, et OMT.**

**1. Le rôle des méthodes:**

La méthode définit un ordre logique dans le quel les tâches doivent être réalisé pour atteindre un but définit. Elle spécifie l'inventaire, la nature et le contenu des tâches ainsi que l'ordre d'exécution et le résultat à l'issue de chaque tâche [RUM 00].

Une méthode est un formalisme qui décompose un processus opératoire en étapes et ces dernières en tâches. Elle représente un fil conducteur qui définit le chemin à suivre pour atteindre assurément, de façon efficace, et sans perdre une cible visé. Ce fil conducteur permet de :

- savoir quoi faire, et dans quelle ordre.
- savoir où on ait réellement, ce qui reste à faire et où on va.

**2. Les différentes classes de méthodes de conception :**

On peut classer les méthodes de conception selon plusieurs critères différents :

- Les étapes du cycle de vie qu'elles supportent.
- La technologie visée (type de langage de programmation, type de SGBD).
- Le type d'applications visées (application de gestion). Cette classification répartit les méthodes en génération :
- Les méthodes d'analyse et de décomposition hiérarchiques qui correspondent à la première génération développée durant les années 70.
- Les méthodes d'analyse et de représentation systémiques qui constituent la deuxième génération des années 80.



-Les méthodes d'analyse et de conception orientées objets qui constituent la troisième génération des années 90.

### **2.1 Les approches cartésiennes (première génération) :**

La première génération inclut les méthodes d'analyse fonctionnelles. Ces méthodes considèrent une fonction à la fois et proposent de la décomposer de façon hiérarchique en un ensemble de sous fonctions jusqu'à atteindre un niveau de décomposition fin pour que le codage des sous fonction soit simple à réaliser.

● Les points forts de cette approche sont :

- La simplicité.
- L'adéquation à capturer plus facilement les besoins des utilisateurs.
- La capacité à produire des solutions à plusieurs niveaux d'abstraction.

● Les points faibles de cette approche :

- Une concentration de l'effort d'analyse sur les fonctions, négligeant la cohérence des données (trop forte redondance).
- Des règles de décomposition non explicites, produisant des hiérarchies de décomposition différentes selon les analystes.
- Une difficulté à tenir compte des interactions non hiérarchiques dans le cas de systèmes complexes.

### **2.2 Les approches systémiques (deuxième génération) :**

Dans ces approches, inspirées de la théorie systémique des organisations, le système d'information est perçu comme un objet complexe actif dont il faut décrire la structure (ou l'architecture) et les objectifs fonctionnels (les fonctions). La modélisation du système est abordée selon deux points de vue complémentaires :

La modélisation des données et la modélisation des traitements. Des modèles de synthèse sont ainsi réalisés pour garantir, d'une part la cohérence des données, et fournir, d'autre part, une spécification la plus complète possible des traitements à réaliser sur ces données.

Les points forts des méthodes basées sur cette approche sont :

- Une plus grande cohérence des données (représentées sous forme d'une base de données non redondante).

*CHAPITRE III*  
*Méthode de conception*  
*et Présentation*  
*de la méthode OMT*



-Le respect des niveaux de conception introduits par le rapport ANSI/SPARC/X3 (Ansi75) (niveau externe, niveau conceptuel et niveau interne).

- La capacité de décrire des systèmes complexes.

Les points faibles de ces méthodes sont :

-Un manque de cohérence entre modèles de données et modèles de traitements (les deux types de modèles n'ont aucun concept commun et ne font pas explicitement référence l'un à l'autre).

-La modélisation des traitements mélange la connaissance et le contrôle (les règles de gestion et les contraintes d'intégrité du système d'information sont intégrées dans la logique algorithmique des fonction, ce qui ne facilite pas leur consultation et leur évolution).

- Un cycle de développement incomplet pour les traitements (ce qui fait dire de ces méthodes que ce sont plus des méthodes de conception que de méthodes de développement).

### **2.3 Les approches objets(troisième génération) :**

On peut considérer l'approche objets comme une évolution de l'approche systémique vers une plus grande cohérence entre les objets et leur dynamique. C'est en effet la principale contribution de l'objet : décrire une grande partie de la dynamique du SI comme un ensemble d'opérations attachées aux objets composant ce système. Ceci donne une meilleure compréhension de la sémantique des objets et permet une meilleure modularité et réutilisation des composants du système d'informations. Comme exemples de méthodes de cette catégorie, on peut citer OOD HOOD, OMT, OOSE, OOA, OOA/OOD.

Les points forts de cette approche résident dans sa capacité à modéliser des objets complexes, à réduire les distorsions entre le réel et système informatique, à exprimer de façon intégrée la dynamique des objets, à encapsuler ces objets pour cacher leur implémentation et ne laisser voir que les services rendus.

Les points faibles sont inhérents à la philosophie du « tout objet » qui pousse à une perception et une représentation monolithique des applications, ce qui ne correspond pas forcément à la perception que l'on a la réalité. L'aspect fonctionnel est très important dans les organisations ; considérer chaque fonction comme un objet relève d'un effort d'abstraction qui n'intéresse guère que les réalisateurs de compilateurs ou les spécialistes dont le but est d'identifier un sous-ensemble minimal de concepts à réaliser. Fort heureusement certaines méthodes, comme



OMT, conservent cette dimension fonctionnelle. Leur succès réside certainement en partie de ce compromis fait avec les méthodes classiques.

### 3. Pourquoi l'OMT ? :

Dans cette étude, on a opté pour la méthode OMT pour les raisons suivantes : [BOU 94]  
C'est l'une des méthodes orientées objets les plus complètes. OMT se définit comme une méthode de modélisation Orientée Objet, organisée autour des concepts du monde réel.

Les trois modèles de cette méthode (Objet, Dynamique, Fonctionnel) traités dans l'ordre du : « Quoi », « Quand », « Comment » ne sont pas indépendants les uns des autres, cette dépendance constitue une forme de validation puisqu'elle permet de modifier les modèles en rajoutant ou substituant des éléments au cours de la modélisation.

De plus, OMT est une extension du modèle entité/ Association utilisée, très proche de celui de MERISE, qui a fait ses preuves tout au long des années passées.

### 4. Présentation de la méthode « OMT » (Object Modeling Technique) :

La technique de modélisation par objet (*Object Modeling Technique OMT*) est une méthode orientée objet de développement de système d'information, développée par Rumbaugh (1991). Elle permet *l'analyse* des spécifications du problème, *la conception* d'une solution au problème et enfin, *l'implémentation* de la solution.

Dans la méthode OMT, trois modèles différents sont employés pour capter un système:

#### 4.1 Le modèle statique :

Le modèle statique de la méthode OMT est une extension du modèle « entité association ».

Le modèle entité association utilisé est très proche de celui de merise, tant ce qui concerne ces concepts que sa représentation graphique. C'est l'une des rares méthodes américaines à accepter la représentation d'association n-aires.

Les extensions apporter sont l'agrégation, généralisation d'une part et la spécification d'opérations sur les objets d'autre part. Chaque entité ou association est vue comme une classe d'objet décrite par une liste d'attribut et une liste d'opérations. [MFW 97]

#### Les diagrammes d'objet :

Propose une notation graphique formelle qui permet de modéliser, les classes et les relations qu'elles entretiennent. Les diagrammes d'objets sont utiles à la fois pour la modélisation abstraite et

pour la conception des programmes réels. Ils offrent l'avantage d'être concis, facile à comprendre et de bien fonctionner dans la pratique.

Ils existent deux types de diagramme d'objet :

Les diagrammes de classe et les diagrammes d'instance.

Un diagramme de classe est un schéma, une structure permet de décrire un grand nombre d'instance possible de donnée. Il décrit les classes d'objet qui décrivent le cas général de modélisation d'un système

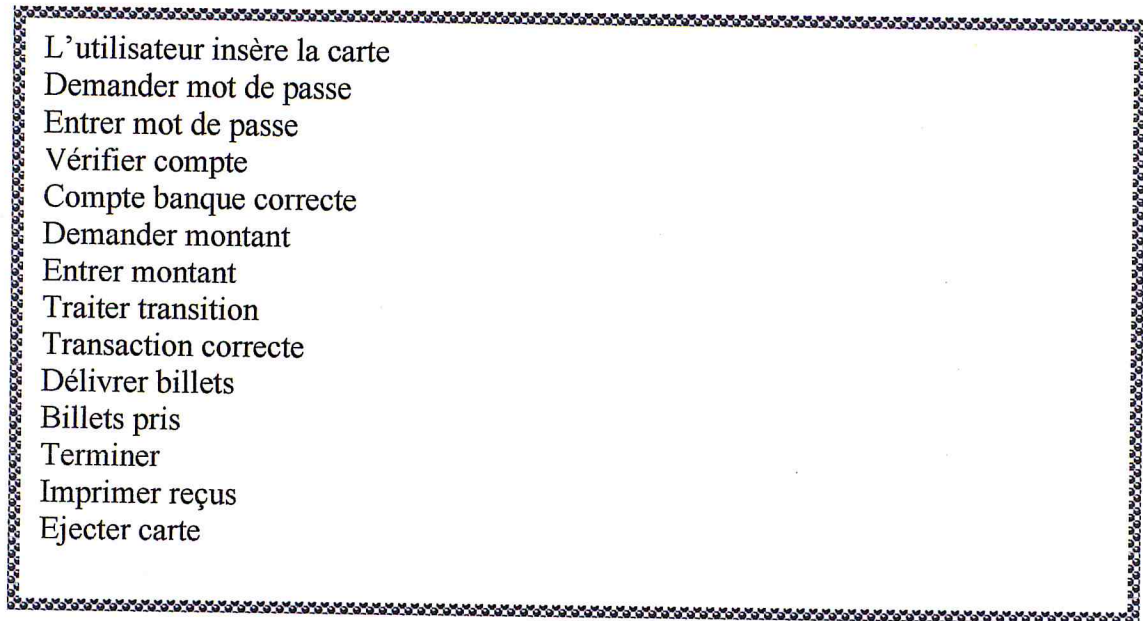
Un diagramme d'instance décrit comment un ensemble particulier d'objet est lié à d'autre. Ces diagrammes sont souvent utiles pour expliquer des cas de test (et tous particulièrement des scénarios) et pour présenter des exemples. A un diagramme de classe donné correspond un ensemble infini de diagramme d'instance. [MFW 97]

#### **4.2 Le modèle dynamique:**

L'objectif de ce modèle est de décrire le cycle de vie des objets (liste d'états possible). Pour cela, OMT utilise des diagramme d'état /transition sur les quelle sont porter les événements déclencheurs des transitions et les opérations de transformation correspondante. Le passage d'un état à un autre se traduit par la modification de la valeur d'un ou de plusieurs attributs qui caractérise cet événement, la condition à vérifier en plus de l'occurrence d'événement et l'action à exécuter. [MFW 97]

##### **4.2.1 Scénario:**

Un scénario est une séquence d'événement se déroulant durant une extension particulière d'un système. La porté d'un scénario peut varier, il peut inclure tout les événement du système ou seulement ce qui entraînent un conflit, ou encore se produit par certains objets du système. Un scénario peut être un enregistrement historique de l'exécution d'un système ou l'expérimentation d'exécution du système proposer [OMT]. La **Figure III.1** montre le scénario de la gestion des transactions bancaires. [MFW 97]

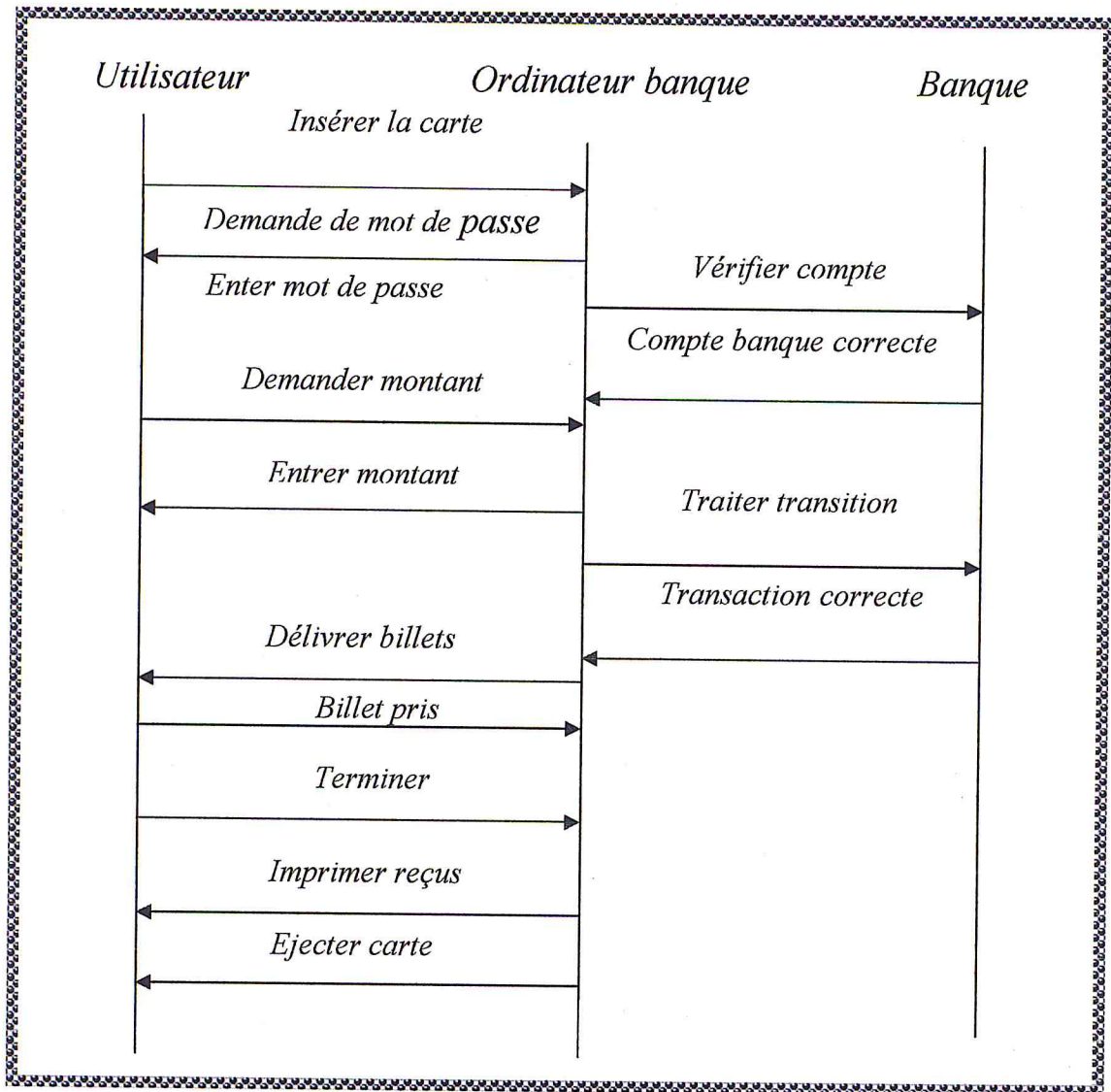


**Figure III.1 Scénario pour la gestion des transactions bancaires**

#### 4.2.2 Diagramme de suivi d'événement:

La première chose à faire après avoir écrit un scénario et d'identifier les objets émetteurs et récepteurs de chaque événement. La séquence d'événement et les objets échangeant ces événements peuvent tous apparaître dans un "scénario étendu" appelée diagramme de suivi d'événement (event trace diagram). Ce diagramme représente chaque objet par une ligne verticale et chaque événement par une flèche horizontal reliant l'objet émetteur à l'objet récepteur. Le temps s'incrémente du haut vers le bas de la figure, mais les espaces ne sont pas significatifs, seul les séquences d'événement sont représenté, non le temps qui les sépare [MPW 97].





**Figure III.2 Diagramme de suivi d'événement pour la gestion des transactions Bancaires**

#### 4.2.3 Diagramme d'état:

Un diagramme d'état relie des événements à des états. Quand un événement est reçu l'état suivant dépend de l'état courant au tant que de l'événement, les diagrammes d'état de l'ensemble des classes constituent le modèle dynamique du système d'informations. Une modification d'état provoquée par un événement est appelée transition. Un diagramme d'état est un graphe dont les nœuds sont des états et les arcs orientés des transitions désignées par les nom d'événements. Un état est représenté par une boîte arrondis contenant un nom optionnel. Une transition est

représenté par une flèche de l'état receveur vers l'état cible, le désignant la flèche est celui de l'événement provoquant la transition. Toute les transitions quittant un état donné doivent correspondre a différent événements.

Dans le diagramme ci-dessous, les principaux événements sont des clics de souris réalisés par l'utilisateur. Les actions accompagnants à chaque événement sont considérées comme atomique. Les activités affichées est masquée introduites par le mot clé «do» sont des activités permanentes. [MFW 97]

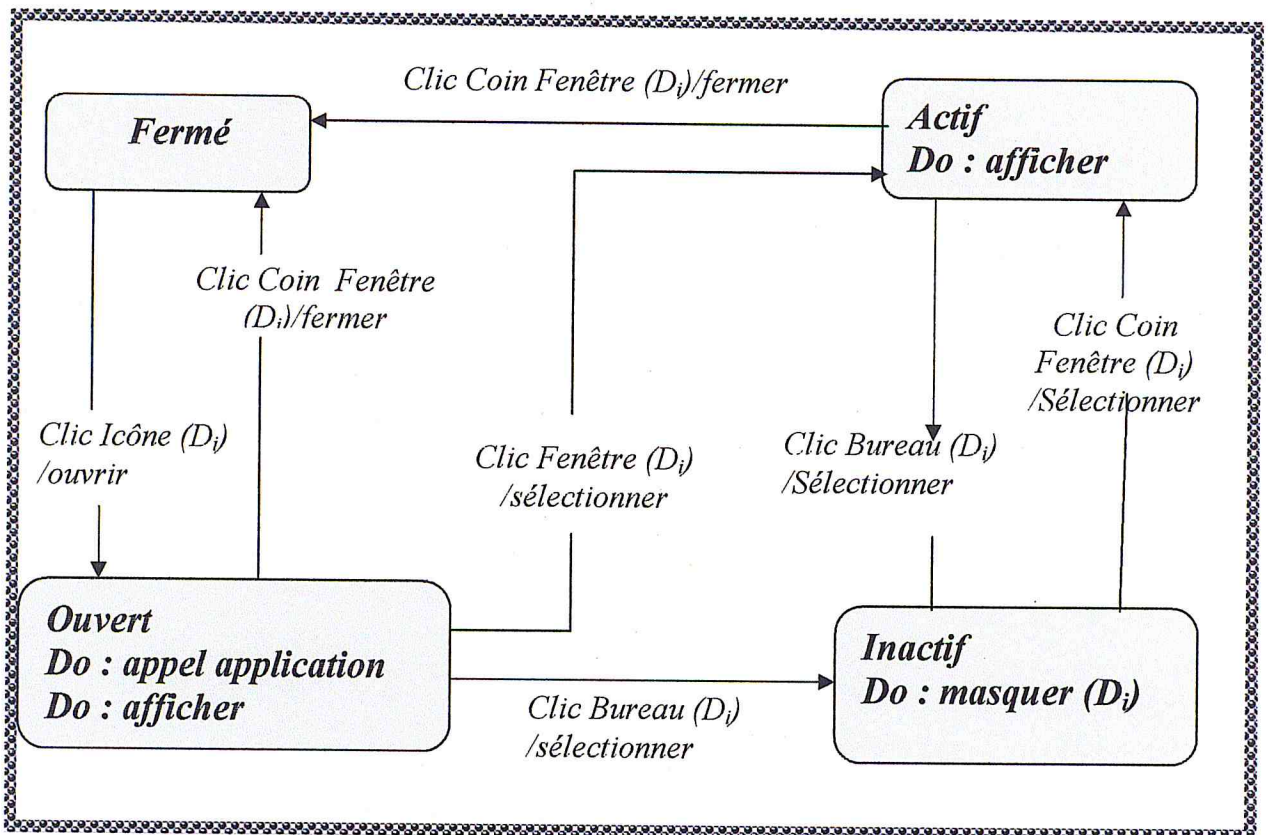


Figure III.3 Exemple de modèle dynamique

#### 4.3 Modèle fonctionnel:

Le model fonctionnel décrit les processus de transformation de l'application (ou les fonctions). Ce processus est une formalisation opérationnelle d'un scénario.

L'OMT utilise le modèle fonctionnel pour décrire les fonctions du système d'information. Le modèle utiliser basé sur les diagrammes de flux de donnée (DFD) classique, comportant des acteurs, des processus, des récipients et des flux de donnée circulant entre ces éléments.

[MFW 97]

**Diagramme à flot de donnée :**

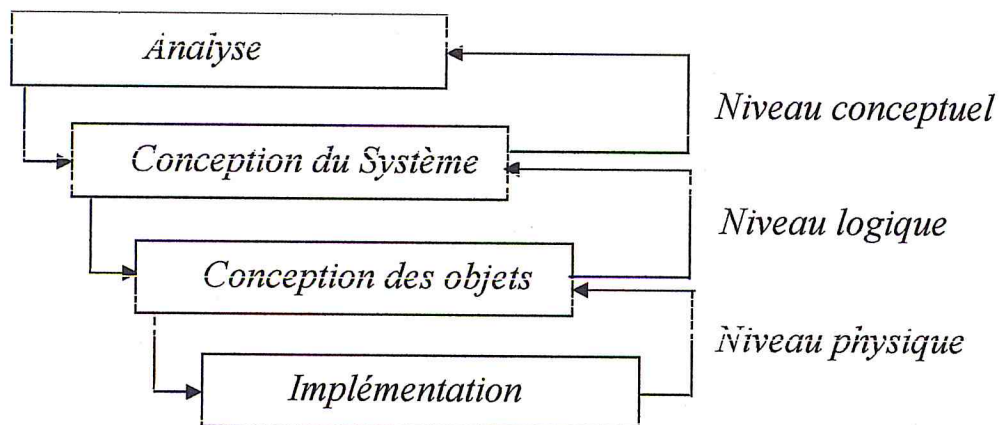
Le modèle fonctionnel consiste en de multiple diagramme a flot de donnée qui précisent la signification des opérations et des contrainte. Un diagramme a flot de donnée (DFD) montre les relations fonctionnelles entre les valeurs calculés par un système, y compris les valeurs entrantes, les valeurs sortantes et les réservoirs de donnée interne. Un diagramme à flot de donnée ne gère pas l'information de contrôle comme le moment ou s'exécute un traitement, ou bien le choix parmi les chemins possible d'accès à la donnée.

Un diagramme à flot de donnée contient des traitements qui transforment les données, des flots de donnée qui transportent les données, des objets acteurs qui produisent et consomment les données, et des objets réservoirs des données qui stockent passivement les données.

[MFW 97]

**5. La démarche méthodologique:**

Le cycle de développement de l'OMT est un cycle en cascade dont lequel les notion de système et de composant sont introduit comme dans le cycle en V.



**Figure III.4 Cycle de développement d'OMT**



### **5.1 L'analyse:**

Le but de cette étape est de développer un modèle de ce que le système doit faire. Ce modèle est exprimé en terme d'objet et de relation de flux dynamiquement de transformation fonctionnelle.

Les discussions avec le client doivent se poursuivre pendant toute l'analyse

1- Ecrire ou obtenir une description initiale du problème (la définition du problème).

2- Construire un modèle objet.

Modèle objet = diagramme du modèle objet + dictionnaire de donnée.

3- Développer un modèle dynamique.

Modèle dynamique = diagramme d'état + diagramme globale des flots d'événements.

4 - Construire un modèle fonctionnel.

Modèle fonctionnel = diagramme à flot de donnée + contrainte.

5 Vérifier, étirer et affiner les trois modèles.

Documentation d'analyse = définition du problème + modèle objet + modèle dynamique + modèle fonctionnel. [MFW 97]

### **5.2 La Conception du système:**

Durant la conception du système le choix de la structure du haut niveau du système est arrêté.

L'approche orienté objet : n'introduit au sein éclairage particulier dans la conception du système, mais nous incluant cette phase dans un souci de couverture complète du cycle de vie.

1- Organiser le système en sous système.

2 -Identifier la concurrence inhérente au problème.

3- Allouer les sous systèmes à des processeurs et à des tâches.

4- Choisir la stratégie de base pour implanter le stockage des données en terme de structure de donnée, de fichier et de base de donnée.

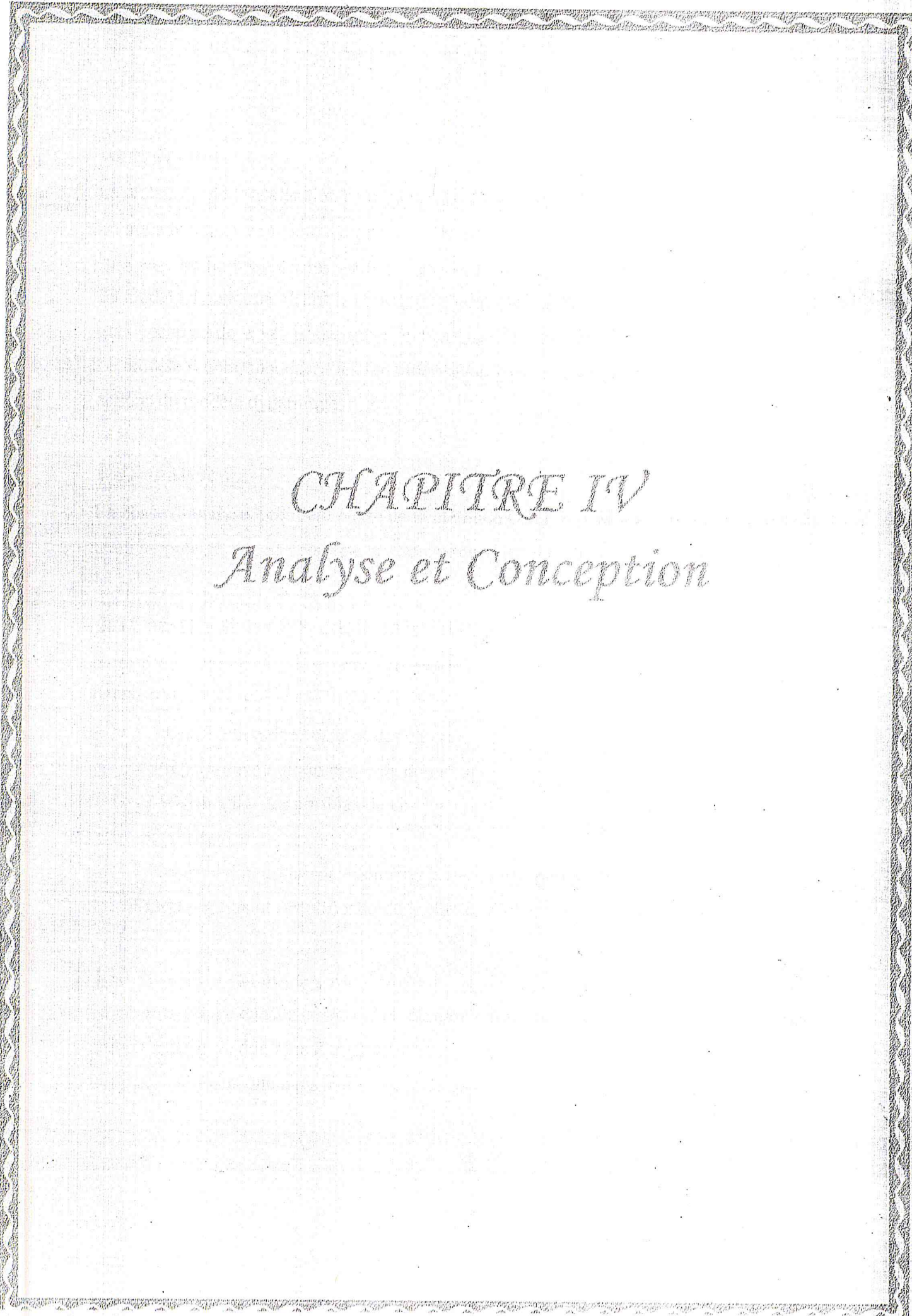
5- Identifier les ressources globales et déterminer les mécanismes de contrôle des accès.

6- Choisir une approche pour l'implémentation du contrôle logiciel.

7- Examiner les conditions critiques.

8 -Etablir les compromis de priorité.

Documentation de conception du système = structure de l'architecture de base du système et décision stratégique de haut niveau. [MFW 97]



CHAPITRE IV  
*Analyse et Conception*



**Introduction :**

La technique de modélisation par objets (OMT) associe trois modèles liés mais distincts :

Le modèle objet, le modèle dynamique et le modèle fonctionnel. Le modèle objet décrit la structure des données sur à laquelle le modèle dynamique et le modèle fonctionnel opèrent.

Le modèle dynamique décrit la structure de contrôle des objets, met en évidence les décisions qui dépendent de la valeur des objets et invoque des fonctions.

Le modèle fonctionnel décrit les fonctions invoquées par les opérations du modèle objet et les actions du modèle dynamique.

**1. L'analyse :**

La phase d'analyse, première phase de la méthode OMT, a pour objectifs de décrire de manière précise, concise correcte et compréhensible un modèle du monde réel.

**1.1 Construction du modèle objet :** le modèle objet fournit la structure des données à partir d'un système réel. Les étapes nécessaires à la construction de ce modèle sont :

[RUM 95].

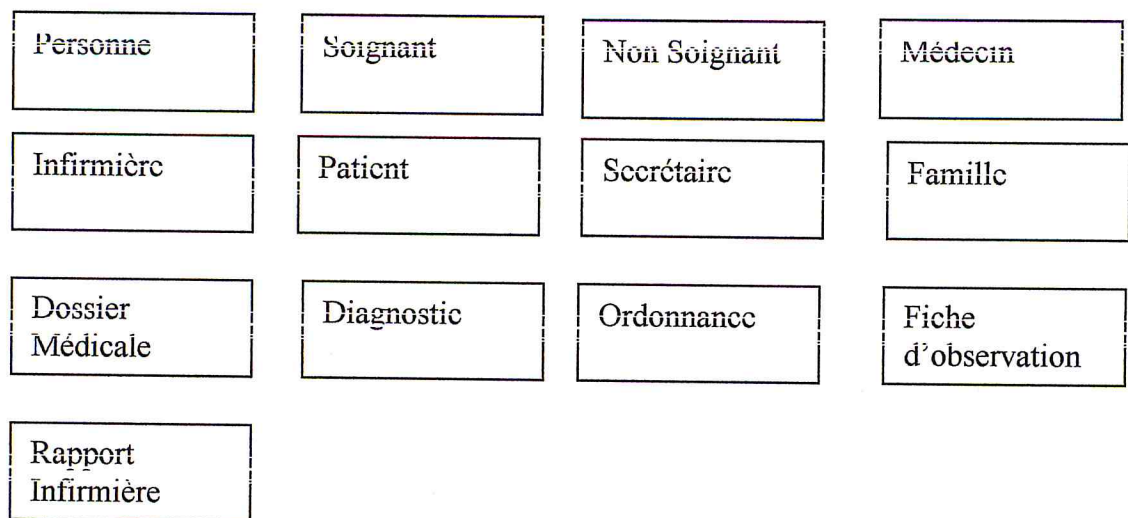
- Identification des objets et des classes.
- Préparation du dictionnaire de données.
- Identification des attributs de chaque objet.
- Codification
- Identification des associations (incluant les agrégations) entre objets
- Organisation et simplification des classes d'objet en utilisant l'héritage.

**1.1.1 Identification des classes d'objets :**

La première étape dans la construction d'un modèle objet consiste à identifier les classes d'objet à partir du domaine d'application.

Les classes d'objets que nous avons recensés dans notre étude sont :





### 1.1.2 Dictionnaire de donnée :

Un dictionnaire de donnée est un paragraphe qui décrit chaque classe d'objet. Il décrit le cadre d'utilisation de la classe dans le problème courant, avec les hypothèses et les restrictions faites sur ses ensembles.

**Personne** : classe qui identifie le personnel de l'hôpital.

**Soignant** : classe qui identifie le personnel soignant.

**Infirmière** : classe qui représente l'ensemble des infirmières.

**Médecin** : classe qui identifie l'ensemble des médecin de l'hôpital.

**Non soignant** : classe qui représente le personnel non soignant.

**Patient** : classe qui identifie les patients (malades).

**Secrétaire** : classe qui représente les secrétaires.

**Famille** : classe qui regroupe la famille du patient.

**Dossier médicale** : classe qui représente les dossiers médicaux des patients.

**Diagnostic** : classe qui identifie le diagnostic de la maladie d'un patient.

**Ordonnance** : classe qui représente les ordonnances délivrées au patient par le médecin.

**Fiche d'observation** : classe qui représente les fiches d'observation d'un patient.

**Rapport infirmière** : classe qui représente le rapport de l'infirmière.

1.1.3 Identification des attributs :

Les attributs sont des propriétés d'un objet, nous avons pu dégager les attributs suivants :

Classe	Attributs	Signification	type	taille
<b>Personne</b>	Mat	Matricule de la personne	N	10
	Nom pers	Nom de la personne	A	15
	Prénom	Prénom de la personne	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08
	Lieu Nais	Lieu de naissance	A	20
	Adr	Adresse de la personne	AN	30
	N°Tel	Numéro de téléphone	N	12
<b>Soignant</b>	Mat Soig	Matricule de la personne	N	10
	Nom	Nom de la personne	A	15
	Prénom	Prénom de la personne	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08
	Lieu -Nais	Lieu de naissance	A	20
	Adr	Adresse de la personne	AN	30
	N°Tel	Numéro de téléphone	N	12
	Poste	Poste de la personne (Chef service)	A	20
	N°Service	Numéro de service	N	08
N°SS	Numéro de sécurité sociale	N	08	
<b>Non Soignant</b>	Mat N Soig	Matricule de la personne	N	10
	Nom	Nom de la personne	A	15
	Prénom	Prénom de la personne	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08
	Lieu -Nais	Lieu de naissance	A	20
	Adr	Adresse de la personne	AN	30
	N°Tel	Numéro de téléphone	N	12
<b>Médecin</b>	Mat Med	Matricule du médecin	N	10
	Nom	Nom du médecin	A	15
	Prénom	Prénom du médecin	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08

	Lieu -Nais	Lieu de naissance	A	20
	Adr	Adresse du médecin	AN	30
	N°Tel	Numéro de téléphone	N	11
	Poste	Poste du médecin (Chef service)	A	20
	N°Service	Numéro de service	N	08
	N°SS	Numéro de sécurité sociale	N	08
	Date- Rec	Date de recrutement du médecin	Date	08
	Grade	Grade du médecin (Professeur, maître assistant)	A	20
	Spécialité	Spécialité du médecin	A	20
	S A	Situation actuelle (Actif ou inactif)	A	10
<b>Infirmière</b>	Mat Inf	Matricule de l'infirmière	N	10
	Nom	Nom de l'infirmière	A	15
	Prénom	Prénom de l'infirmière	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08
	Lieu -Nais	Lieu de naissance	A	20
	Adr	Adresse de l'infirmière	AN	30
	N°Tel	Numéro de téléphone	N	12
	Poste	Poste de l'infirmière (Chef service)	A	20
	N°Service	Numéro de service		
	N°SS	Numéro de sécurité sociale	N	08
	Date- Rec	Date de recrutement	N	08
	Spécialité	spécialité de l'infirmière (infirmière de soin, de laboratoire, kinésithérapeute)	Date	08
			A	20
<b>Patient</b>	Mat Pat	Matricule du patient	N	10
	Nom	Nom du patient	A	15
	Prénom	Prénom du patient	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08
	Lieu -Nais	Lieu de naissance	AN	10
	Adr	Adresse du patient	AN	30
	N°Tel	Numéro de téléphone	N	12
	Date -E	Date d'entrée du patient	Date	08
	Age	L'age du patient	N	10
	N°SS	Numéro de sécurité sociale	N	08
	Nom-P	Nom de père du patient	A	20
	Nom-M	Nom de la mère du patient	A	20
	Nationalité	Nationalité du patient	A	20
	Profession	Profession du patient	A	20
	N°Service	Numéro de service	N	10
	N°chambre	Numéro de chambre	N	10
	N°lit	Numéro du lit	N	10
	Existence	Existence du patient dans l'hôpital	A	10



<b>Secrétaire</b>	Mat Sec	Matricule de la secrétaire	N	10
	Nom	Nom de la secrétaire	A	15
	Prénom	Prénom de la secrétaire	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08
	Lieu -Nais	Lieu de naissance	A	20
	Adr	Adresse de la secrétaire	AN	30
	N°Tel	Numéro de téléphone	N	12
	N°SS	Numéro de sécurité sociale	N	08
	Date- Rec	Date de recrutement	Date	08
<b>Famille</b>	Mat Fam	Matricule de la famille	N	10
	Nom	Nom de la famille	A	15
	Prénom	Prénom de la famille	A	20
	Sexe	Sexe	A	09
	SF	Situation Familiale	A	01
	Date-Nais	Date Naissance	Date	08
	Lieu -Nais	Lieu de naissance	A	20
	Adr	Adresse de la famille	AN	30
	N°Tel	Numéro de téléphone	N	12
	Lien de parenté	Lien de parenté (père, proche, mari (e))	A	20
	Nationalité	Nationalité de la famille	A	20
Profession	Profession			
<b>Dossier Médicale</b>	N°Dos	Numéro du dossier	N	20
	Nom-P	Nom du patient	A	20
	Prénom-P	Prénom du patient	A	30
	Date-Naiss	Date de naissance	Date	Date
	Lieu-Naiss	Lieu de naissance	A	20
	Groupe sanguin	Groupe sanguin	A	10
	Date-E	Date d'entrée à L'hôpital pour la première fois	Date	08
<b>Diagnostic</b>	N°Diag	Numéro du diagnostic	N	20
	Nom Maladie	Nom de la Maladie	A	20
	Date diagnostic	Date diagnostic	Date	08
	Nom -médecin signant	Nom du médecin soignant	A	30
	Service	Service	N	20
	Analyses et radio	Analyses et radio	A	30
	Description	Description de la maladie	A	100
	R-Cont	Risque de contamination	A	10
	Type	Type du diagnostic	A	20
	Mode d'entrée	Mode d'entrée (hospitalisation, consultation, urgence)	A	30

<b>Ordonnance</b>	N°Ord	Numéro ordonnance	N	20
	Nom-p	Nom de la personne	A	20
	Prénom-P	Prénom de la personne	A	30
	N-M-S	Nom du Médecin Soignant	A	20
	Sign-M	Signature du Médecin	A	20
	Date ordonnance	Date d'écriture ordonnance	Date	08
	Medicament	Médicament	A	30
<b>Rapport Infirmière</b>	N°Rap	Numéro de rapport Infirmière	A	20
	Nom-P	Nom du patient	A	30
	Prénom- P	Prénom du patient	N	10
	Tension	Tension du patient	N	10
	Poux	Poux du patient	N	10
	Température	Température	N	10
	Dextro	Dextro	N	10
	Apstixe	Apstixe	N	10
	Date du rapport	Date d'écriture du rapport	Date	08
	Nom infirmière	Nom infirmière qui écrit le rapport	N	20
<b>Fiche d'observation</b>	N°Fic	Numéro de la fiche d'observation	A	20
	Nom-P	Nom du patient	A	30
	Prénom-P	Prénom du patient	A	30
	Décision-M	Décision du Médecin	A	30
	Date de fiche	Date d'écriture de la fiche d'observation	Date	08

#### 1.1.4 Codification:

La codification consiste à définir des symboles de conventionnels pour être designer d'une manière mieux adaptée aux traitements automatiques.

##### Les principales qualités de la codification:

- Un code ne doit pas comporter d'ambiguïté.
- Un code doit être conçu de façon à ce que le minimum d'erreur puisse arrivé.
- Extensible (possibilité d'insertion).
- Adapter à l'utilisateur.

##### Avantage de la codification :

###### a. Avantage organisationnel :

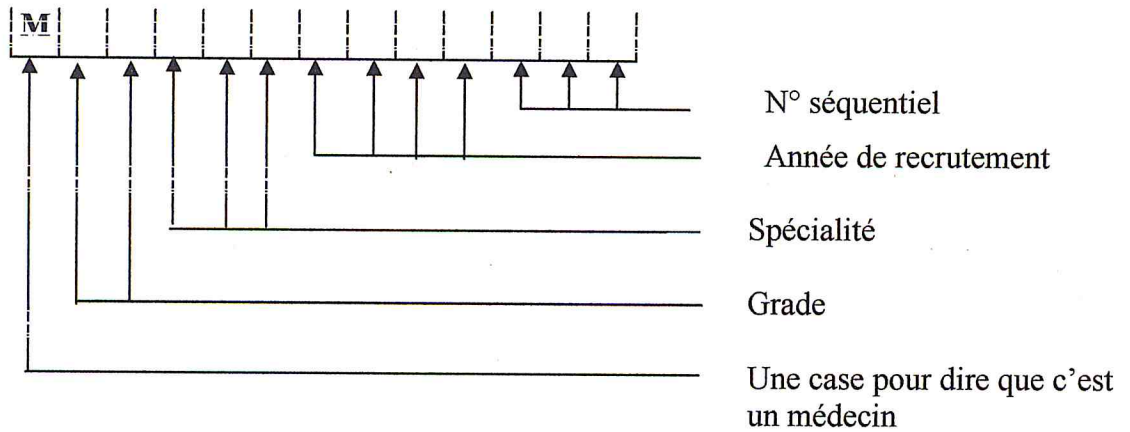
- Faciliter le classement où trient des documents
- Faciliter la recherche ou la consultation des informations.
- Minimisation des erreurs d'enregistrement.

**b. Avantage d'automatisation:**

- consultation rapide et facile.
- détection des erreurs de codification automatique.

**Médecin :**

Code articulé



**Grade de médecin :**

<b>P</b>	Professeur
<b>M</b>	Maître assistant
<b>G</b>	Généraliste

**Spécialité :**

<b>ORL</b>	ORL
<b>CAR</b>	Cardiologie
<b>PNE</b>	Pneumatologie
<b>GYN</b>	Gynécologie
<b>NER</b>	Neurochirurgie

**Exemple :**

La codification d'un médecin professeur en neurochirurgie recruté dans l'année 1986 doter d'un numéro séquentiel 024 est représenté comme suit :

**M P NER 1986 024**

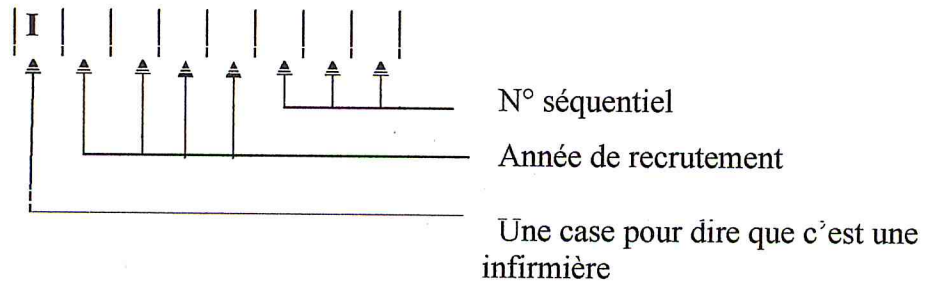
La codification d'un médecin Maître assistant en ophtalmologie recruté dans l'année 2000 doter d'un numéro séquentiel 045 est représenté comme suit :

**M M OPH 2000 045**



**Infirmière :**

Code articulé



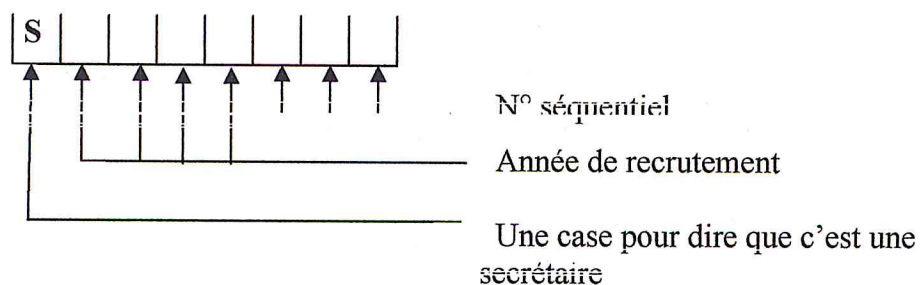
**Exemple :**

La codification d'une infirmière recruté dans l'année 2001 doter d'un numéro séquentiel 059 est représenté comme suit :

**I 2001 059**

**Secrétaire :**

Code articulé



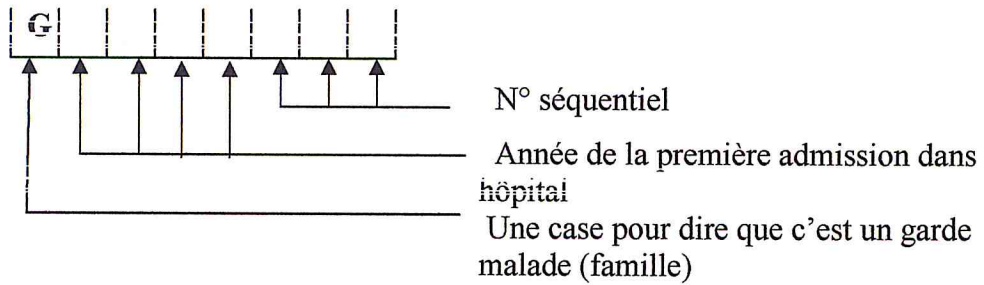
**Exemple :**

La codification d'une Secrétaire recruté dans l'année 2001 doter d'un numéro séquentiel 025 est représenté comme suit :

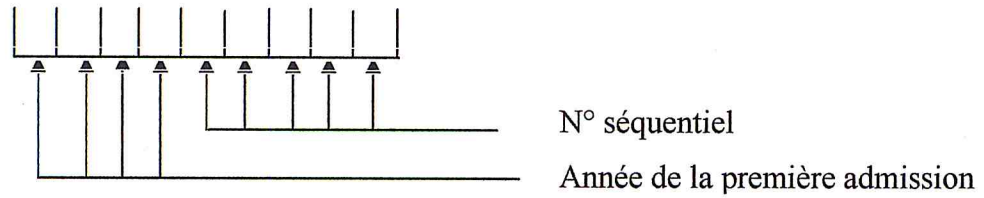
**S 2001 025**

**Famille :**

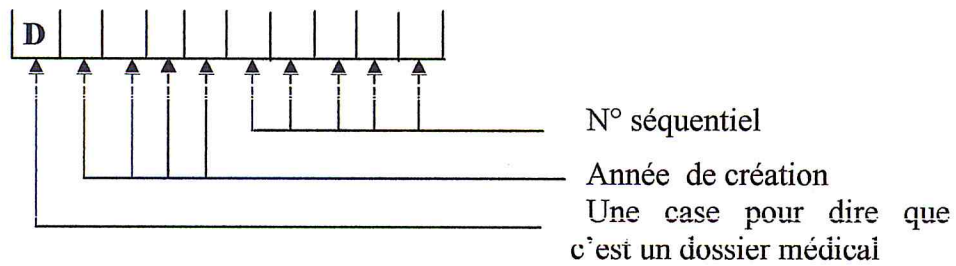
Code articulé



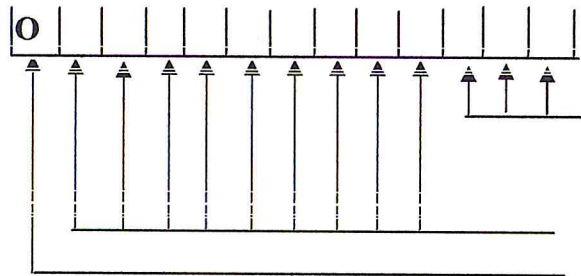
**Patient:**



**Dossier médical:**



**Ordonnance:**

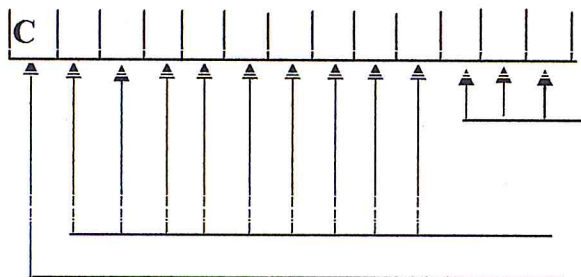


Numéro séquentiel de l'ordonnance

Matricule du patient

Une case pour dire que c'est une ordonnance

**Diagnostic:**

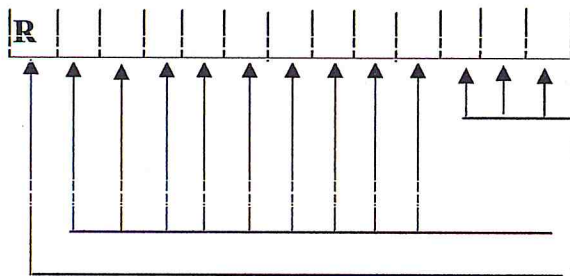


Numéro séquentiel du diagnostic

Matricule du patient

Une case pour dire que c'est un diagnostic

**Rapport infirmière:**



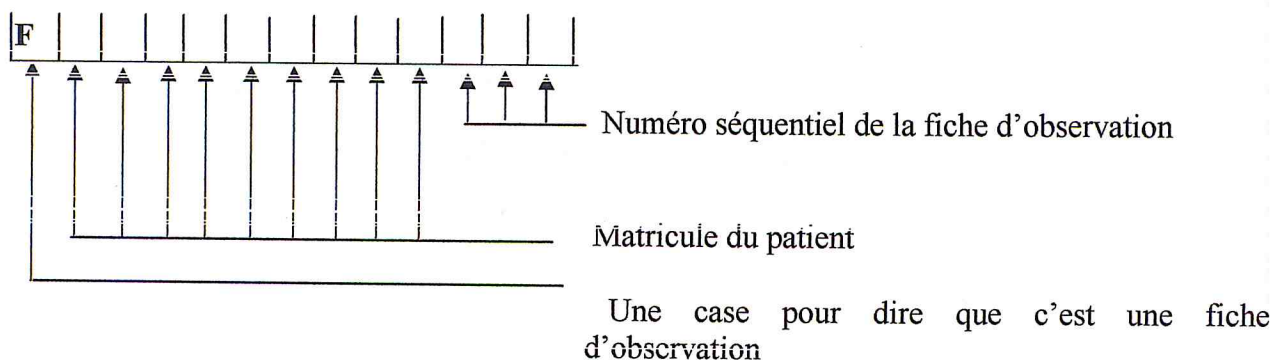
Numéro séquentiel du rapport infirmière

Matricule du patient

Une case pour dire que c'est un rapport infirmière



**Fiche d'observation**



**1.1.5 Identification des associations :**

Toute dépendance entre deux ou plusieurs objets est une association. Dans ce domaine de gestion, la définition de ces associations se détermine par les objectifs et les règles de gestion définis au sein d'une organisation.

Association	Dimension	Cardinalité	Objet intervenant	Propriété de l'association
Donner soins	2	plusieurs à plusieurs	Médecin Patient	
Rédiger	2	Un à plusieurs	Infirmière Rapport Infirmière	
Faire intervention	2	Plusieurs à plusieurs	Médecin Patient	consentement
Avoir	2	Un à un	Patient Dossiers Médical	
Est-Composée-1	2	Un à plusieurs	Dossier Médical Diagnostic	
Est-Composée-2	2	Un à plusieurs	Dossier Médical Fiche d'observations	
Est-Composée-3	2	Un à plusieurs	Dossier Médical Ordonnance	
Est-Composée-4	2	Un à plusieurs	Dossier Médical Rapport Infirmière	

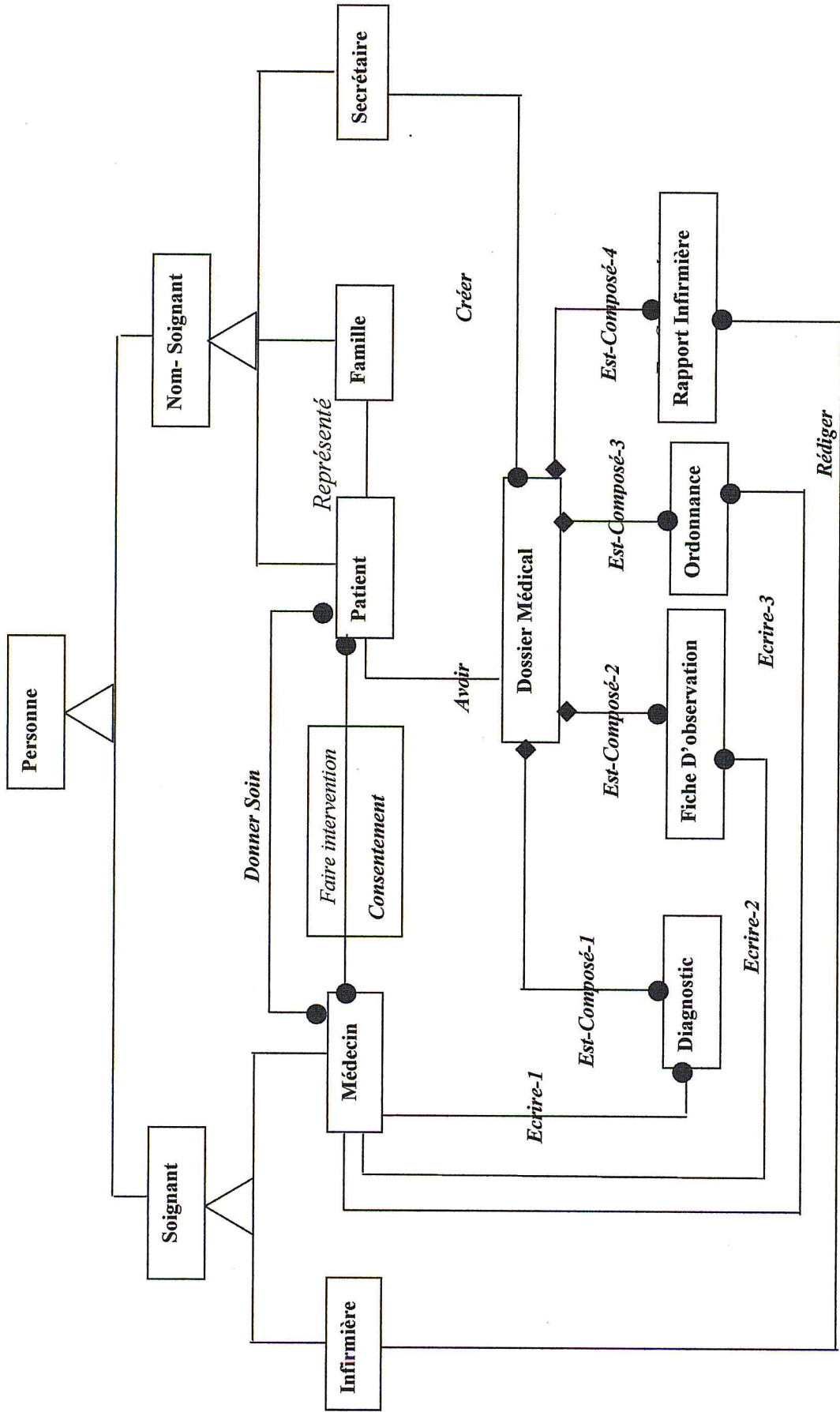


Figure IV.1 Diagramme de classe

## 1.2. Construction du modèle dynamique :

Le modèle dynamique décrit des concepts traitant du flot de contrôle des interactions et des séquences d'opération dans un système d'objet actif (concurrents) [RUM95]

Nous rappelons les deux concepts suivants :

**Événement** : un événement est un fait qui se produit à un moment donné dans le temps, une voie de transmission d'information à sens unique d'un objet vers un autre .

**Scénario** : c'est une séquence d'événement se déroulant durant une exécution particulière d'un système. Chaque événement transmet une information d'un objet à un autre.

Nous préparons les scénarios sous forme de dialogues, ça ne concerne que les classes ayant un aspect dynamique dans le temps, dans le cadre de notre étude les scénarios que nous pouvons identifier sont :

- création du dossier médical.
- Remplir le diagnostic avec un diagnostic non sérieux
- Remplir le diagnostic avec un diagnostic sérieux.
- Remplir le diagnostic avec un diagnostic sérieux et risque de contamination.
- Ecrire rapport infirmière.
- Demande d'examination par la famille et donner soin.
- Demande d'examination par le patient et donner soin.
- Délivrer médicament par le médecin.
- Délivrer médicament par le médecin dans le cas d'une maladie vénérienne ou en cas d'urgence
- Faire intervention.
- Faire intervention dans le cas d'une urgence.

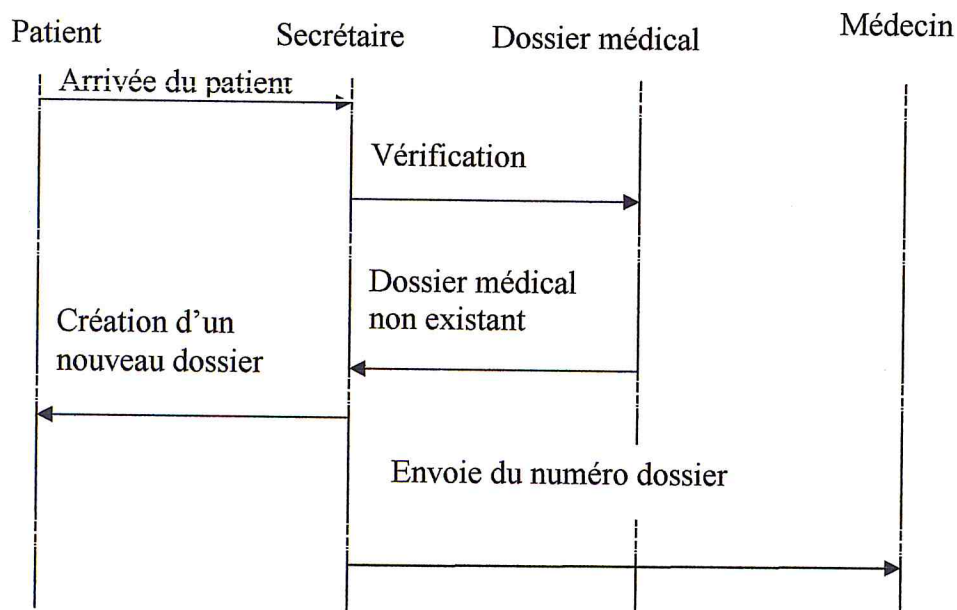


### 1.2.1 Scénarios et suivi d'événements :

#### Scénario création d'un dossier médical

- Arrivée du patient.
- Secrétaire vérifie l'existence du dossier médical.
- Dossier médical non existant.
- Secrétaire crée un nouveau dossier médical
- La secrétaire envoie le numéro du dossier au médecin.

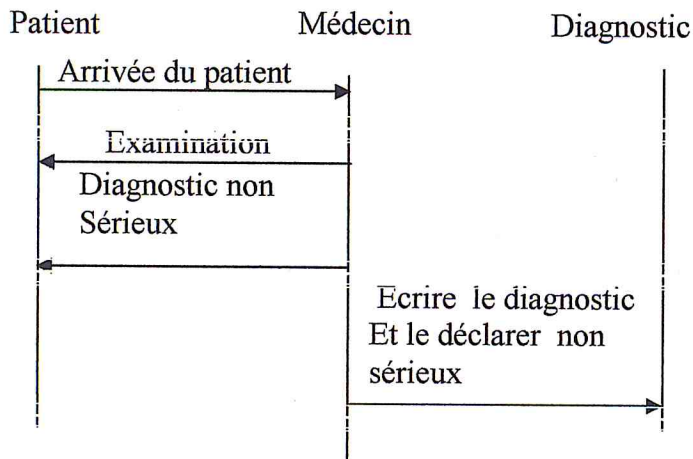
#### Suivi d'événement :



#### Scénario remplir diagnostic avec un diagnostic non sérieux :

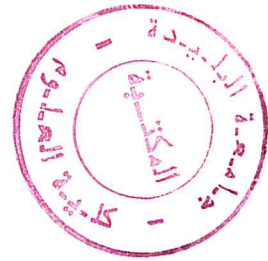
- Arrivée du patient.
- Le médecin examine le patient.
- Si le diagnostic est non sérieux.
- Le médecin écrit le diagnostic et le déclare non sérieux.

Suivis d'événement :

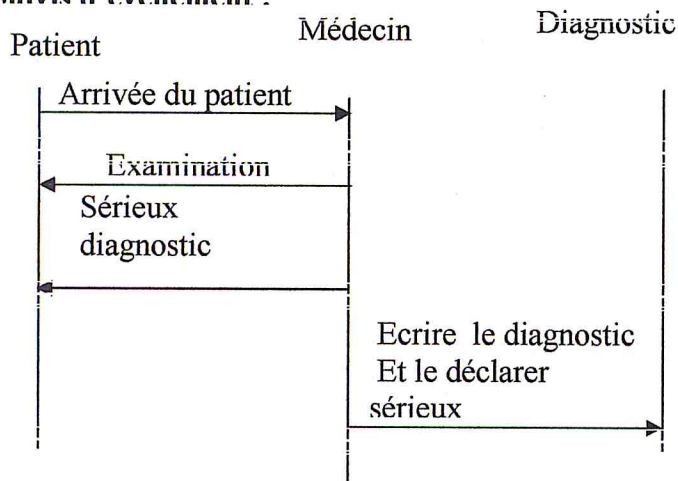


Scénario remplir diagnostic avec un diagnostic sérieux :

- Arrivée du patient.
- Le médecin examine le patient.
- Si le diagnostic est sérieux.
- Le médecin écrit le diagnostic et le déclare sérieux.



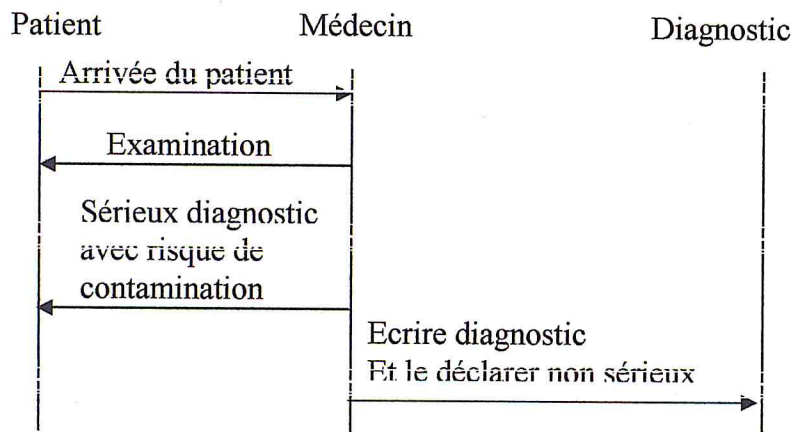
Suivis d'événement :



**Scénario remplir dossier médical avec un diagnostic sérieux et risque de contamination :**

- Arrivée du patient.
- Le médecin examine le patient.
- Si le diagnostic est sérieux avec risque de contamination.
- Le médecin écrit le diagnostic et le déclare non sérieux.

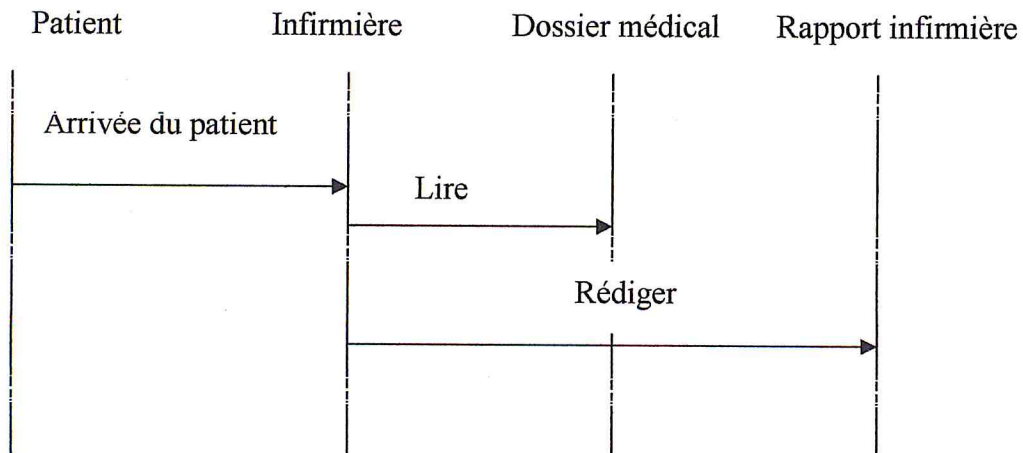
**Suivis d'événement :**



**Scénario écrire rapport infirmière**

- Arrivée du patient.
- L'infirmière lit le dossier médical du patient.
- L'infirmière rédige le rapport infirmière.

**Suivis d'événement :**

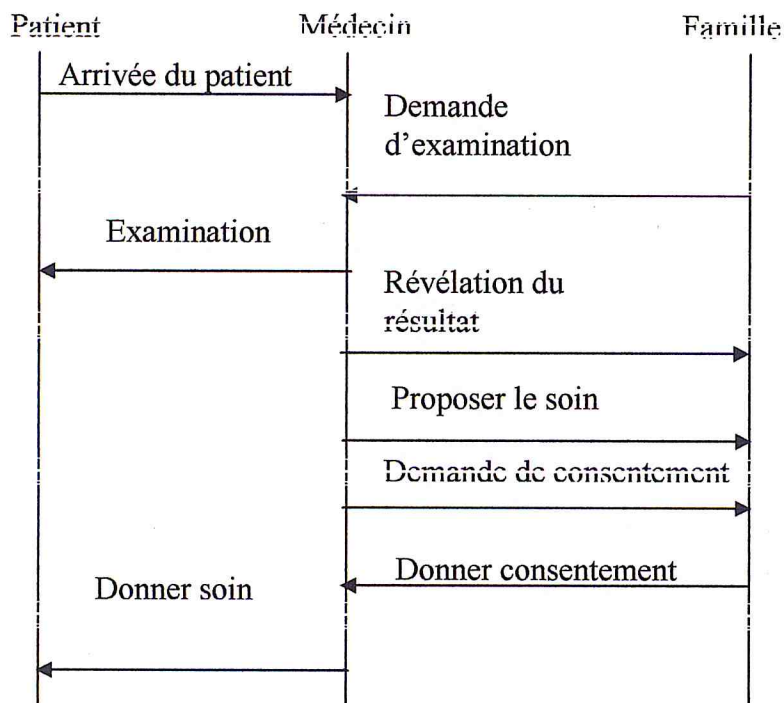




**Scénario demande d'examination par la famille et donner soin**

- Arrivée du patient.
- La famille demande l'examination du patient.
- Le médecin examine le patient.
- Le médecin révèle le résultat d'examen au parent.
- Le médecin propose le soin.
- Le médecin demande le consentement sur le soin à la famille.
- La famille donne le consentement.
- Le médecin donne le soin au patient.

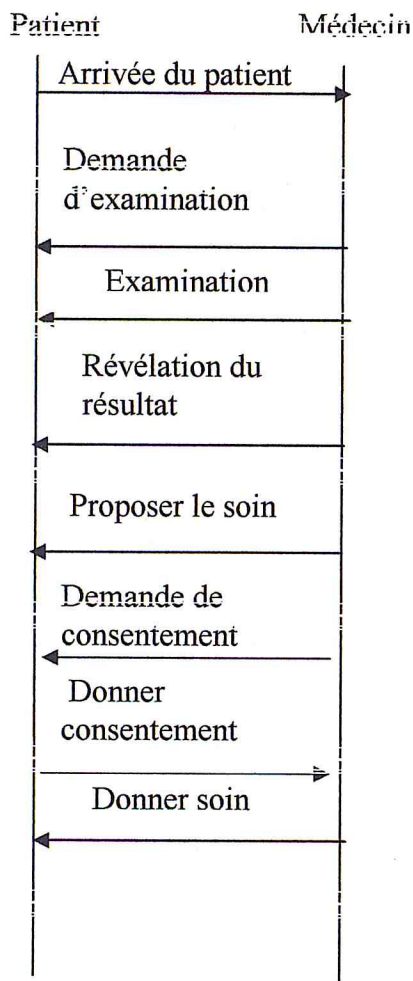
**Suivis d'événement :**



**Scénario demande d'examination par le patient et donner soin**

- Arrivée du patient.
- le patient demande l'examination du patient.
- Le médecin examine le patient.
- Le médecin révèle le résultat d'examen au patient
- Le médecin propose le soin.
- Le médecin demande le consentement sur le soin au patient lui-même.
- Le patient donne le consentement.
- Le médecin donne le soin au patient.

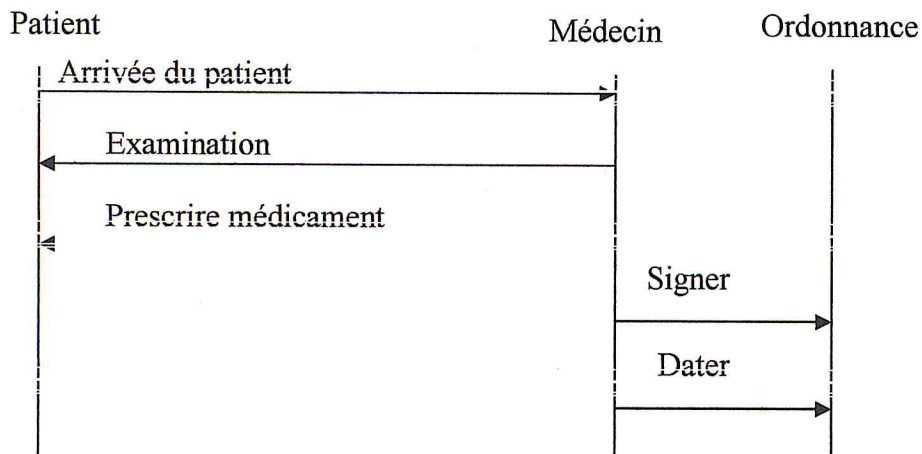
**Suivis d'événement :**



**Scénario prescrire médicament par le médecin**

- Arrivée du patient.
- Le médecin examine le patient.
- Le médecin prescrit médicament.
- Le médecin signe l'ordonnance.
- Le médecin date l'ordonnance.

Suivis d'événement :

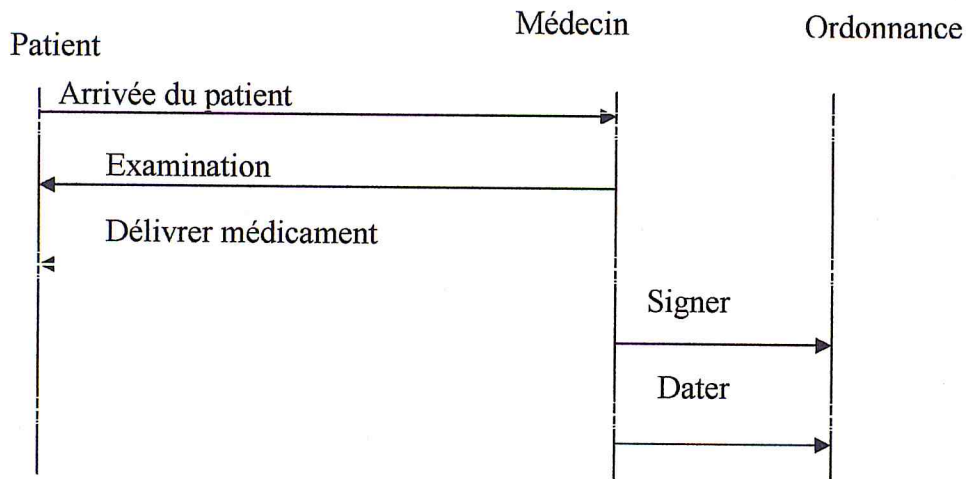


**Scénario délivrer médicament par le médecin dans le cas d'une maladie vénérienne ou cas d'urgence**

- Arrivée du patient.
- Le médecin examine le patient.
- Le médecin délivre les médicaments.
- Le médecin signe l'ordonnance.
- Le médecin date l'ordonnance.



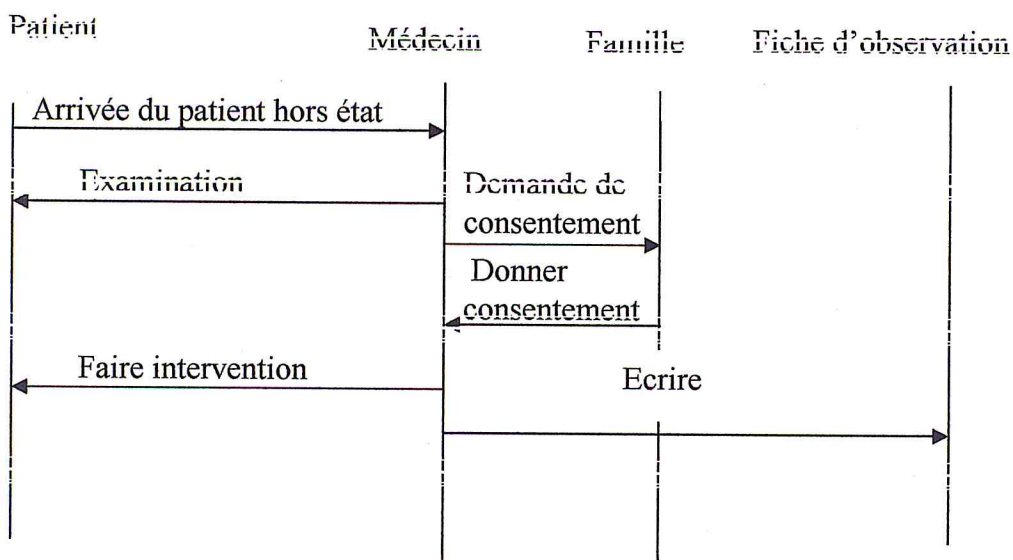
**Suivis d'événement :**



**Scénario faire intervention**

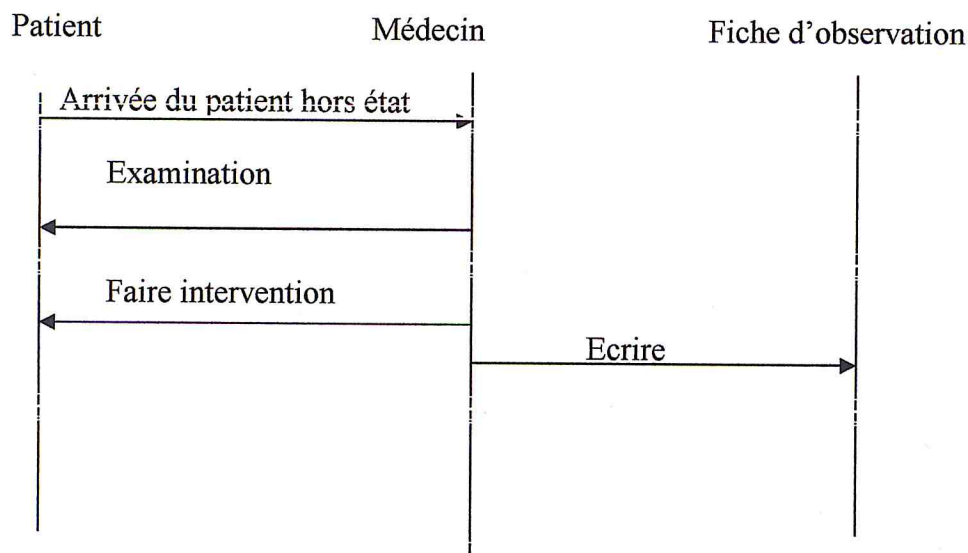
- Arrivée d'un patient hors état.
- Le médecin examine le patient.
- Le médecin demande le consentement à la famille.
- La famille donne son accord
- Le médecin fait l'intervention.
- Le médecin écrit la fiche d'observation.

**Suivis d'événement :**



**Scénario faire intervention dans le cas d'une urgence**

- Arrivée d'un patient hors état.
- Le médecin examine le patient.
- Le médecin fait l'intervention.
- Le médecin écrit la fiche d'observation.

**Suivis d'événement :****1.2.2 Les diagrammes d'états :**

C'est un graphe dont les nœuds sont les états et les arcs orientés des transitions désigner par les noms d'événement.

Pour la représentation des diagrammes nous devons définir ce qui suit :

**État :** c'est une abstraction des valeurs des attributs et des lien d'un objet, il spécifie la réponse de l'objets au événement d'entrée

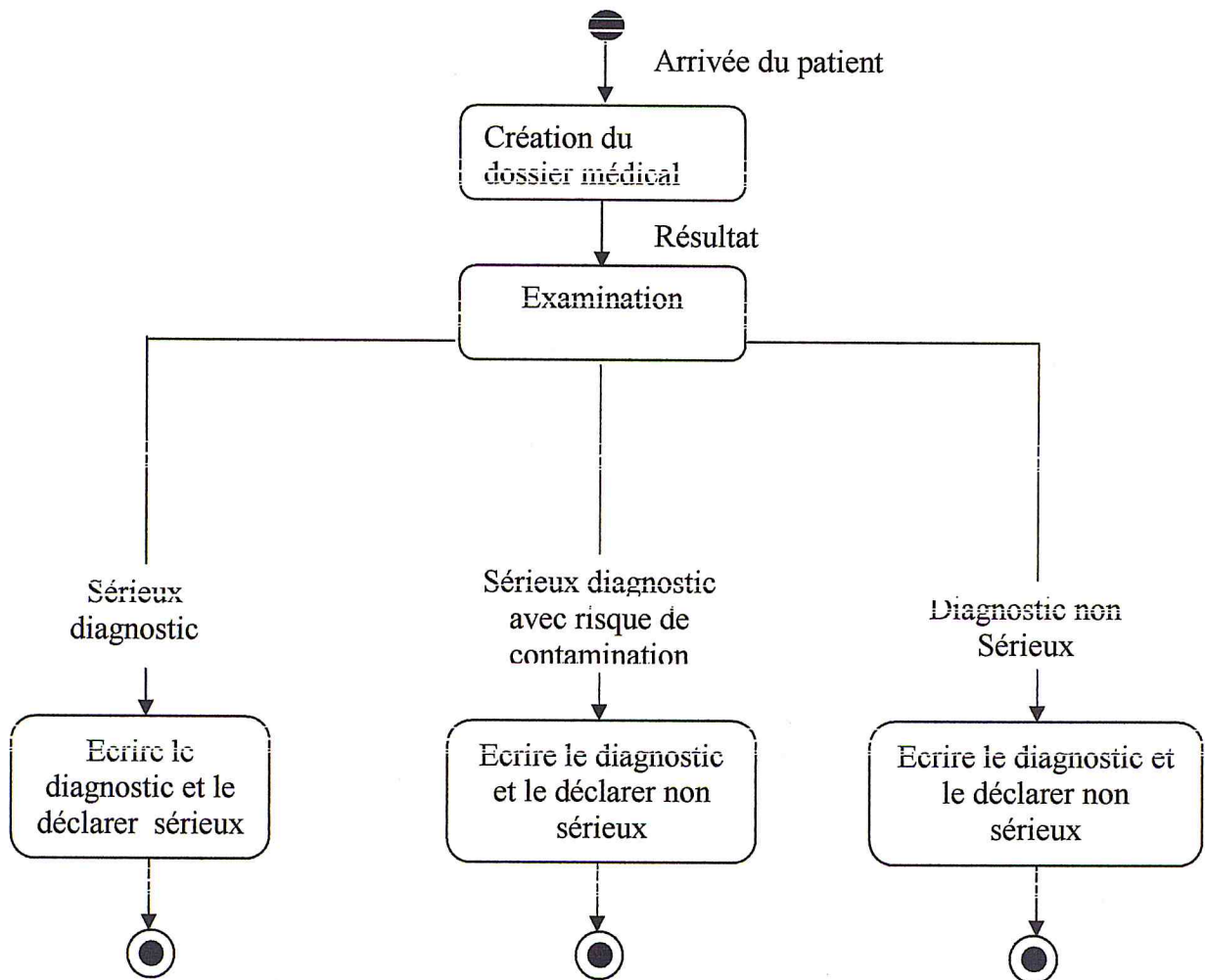
**Les conditions :** une condition est une fonction booléenne utilisé comme grade sur des transitions, elle est représentée entre crochée suivant le nom d'événement.

**Les opérations :** c'est des réponse à des événements (action) ou à des états (activité).

L'activité est représenté par la notation « Faire : A », l'action représentée par une oblique « / » et le nom de l'action suivant le nom de l'événement.

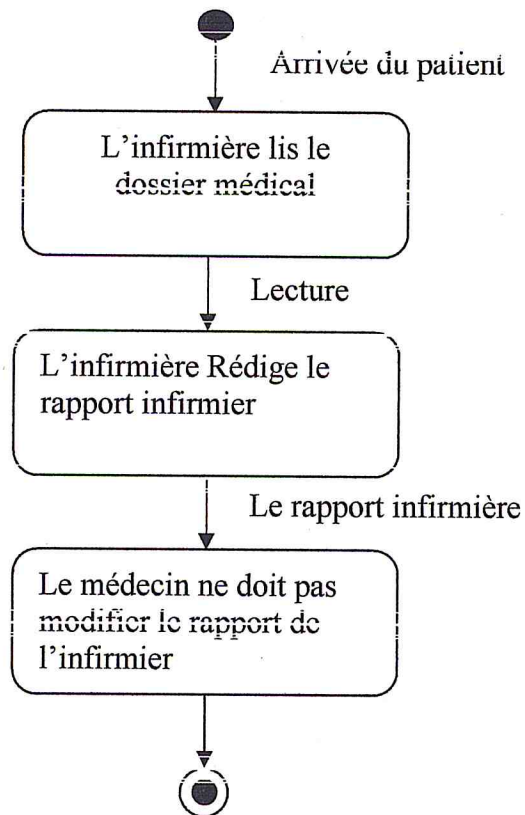
Les diagrammes d'états de notre étude seront les suivants :

**1. diagramme d'état création du dossier médical et écriture du diagnostic**

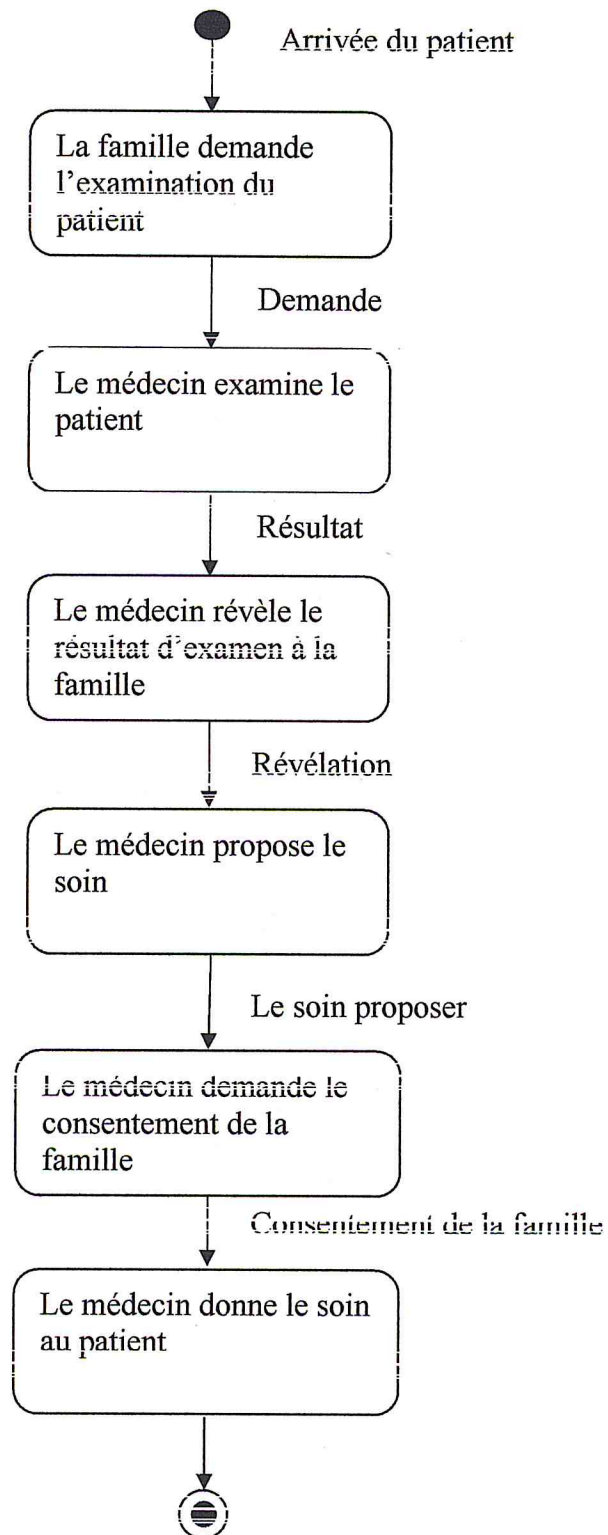




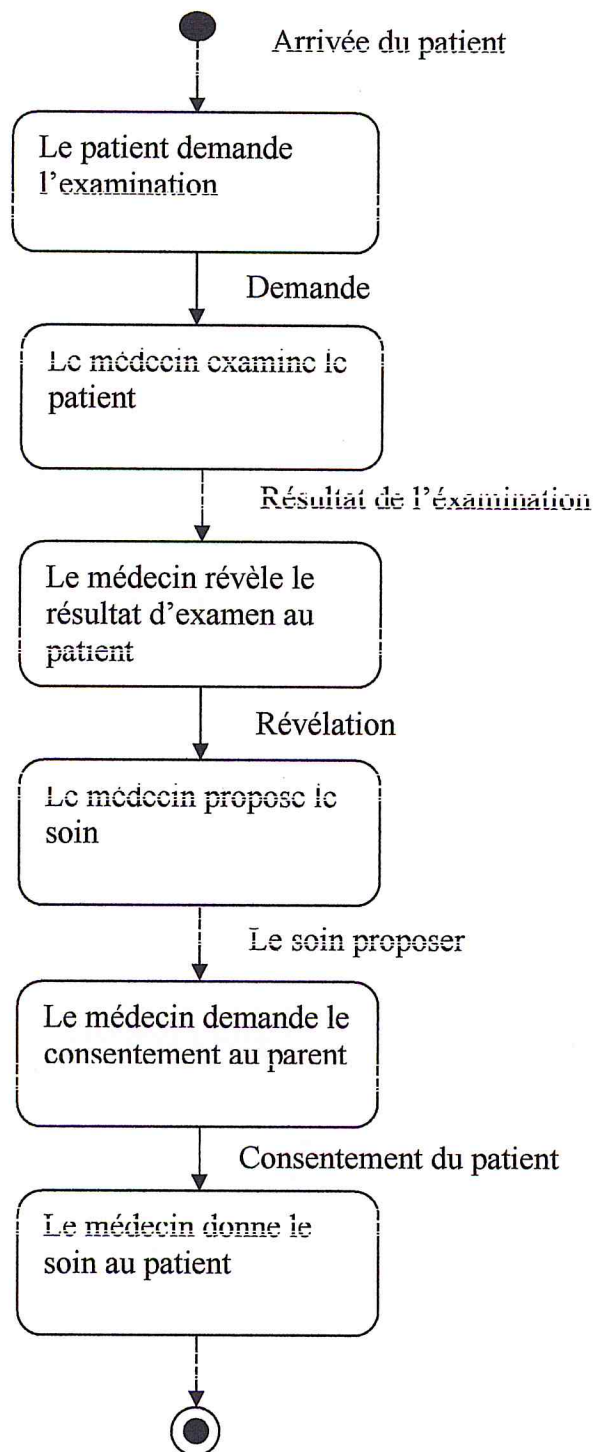
2. Diagramme écrire rapport infirmière



3. Diagramme demande d'examination par la famille et donner soin

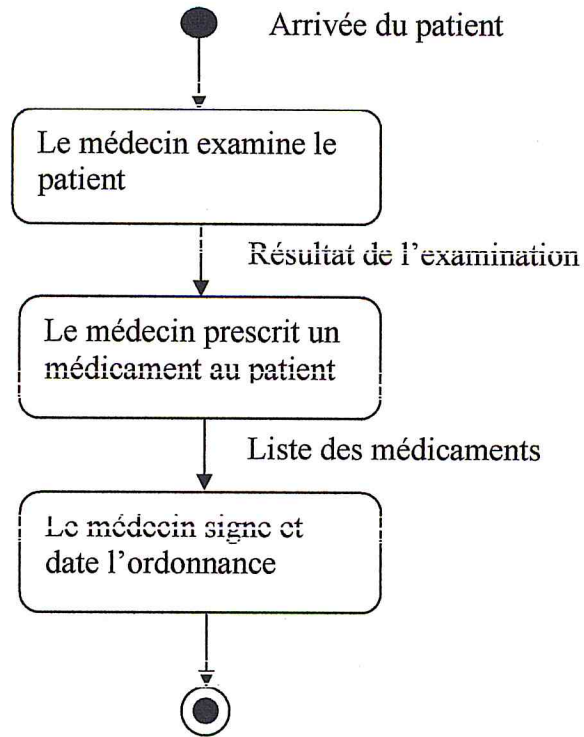


4. Diagramme demande d'examination par le patient et donner soin

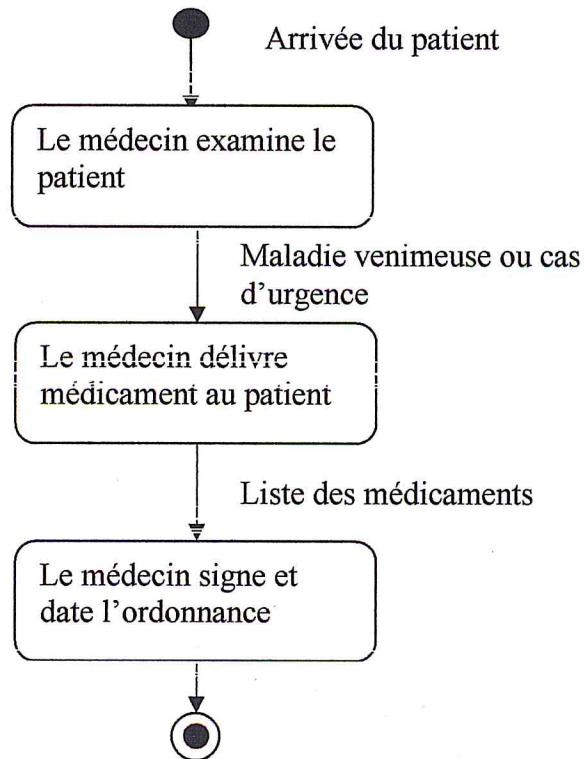




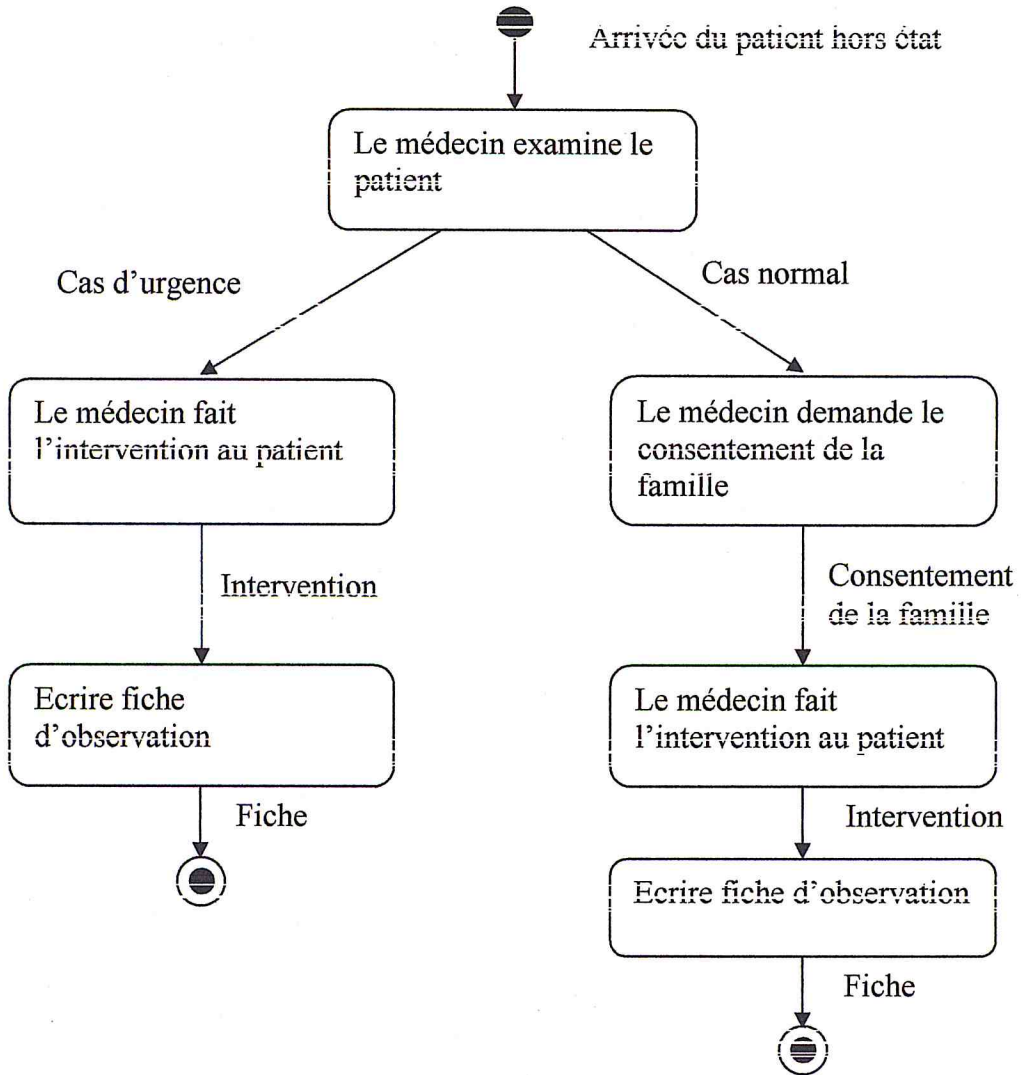
**5. Diagramme d'état prescrire médicament**



**6. Diagramme d'état délivrer médicament**



7. Diagramme d'état faire une intervention



### 1.3 Modèle fonctionnel :

Le modèle fonctionnel indique les résultats d'un calcul sans préciser quand et comment ils ont été obtenus, il montre comment les valeurs sortantes d'un calcul sont dérivées à partir de valeurs entrantes, il spécifie la signification des opérations dans le modèle objet et la signification des actions dans le modèle dynamique.

Il consiste en plusieurs diagrammes à flots de données.

#### Diagramme à flot de données :

Le DFD contient des traitements qui transforment les données, des flots de données qui transportent des données, des objets acteurs qui produisent et consomment les données et des réservoirs de données qui stockent passivement les données.

Nous aurons à définir les deux notions suivantes :

#### Fonction :

Une fonction est une description concernant les traitements dans les DFD, elle peut être faite en langage naturel ou par des équations mathématique.

#### Contrainte :

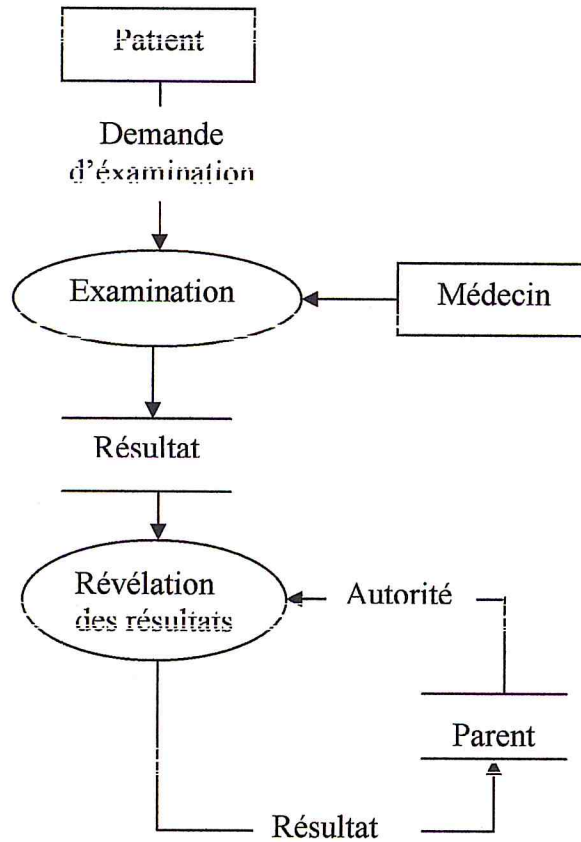
Une contrainte fait apparaître la relation entre deux objets à un moment ou entre valeurs différentes d'un même objet à des moments différents. Elle indique les restrictions sur les opérations.

Les DFD correspondants à notre système seront les suivants :

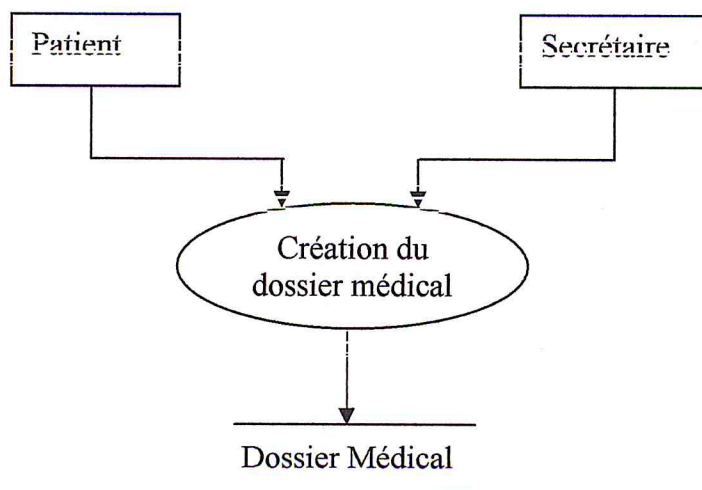


1.3.1 Diagramme à flot de données :

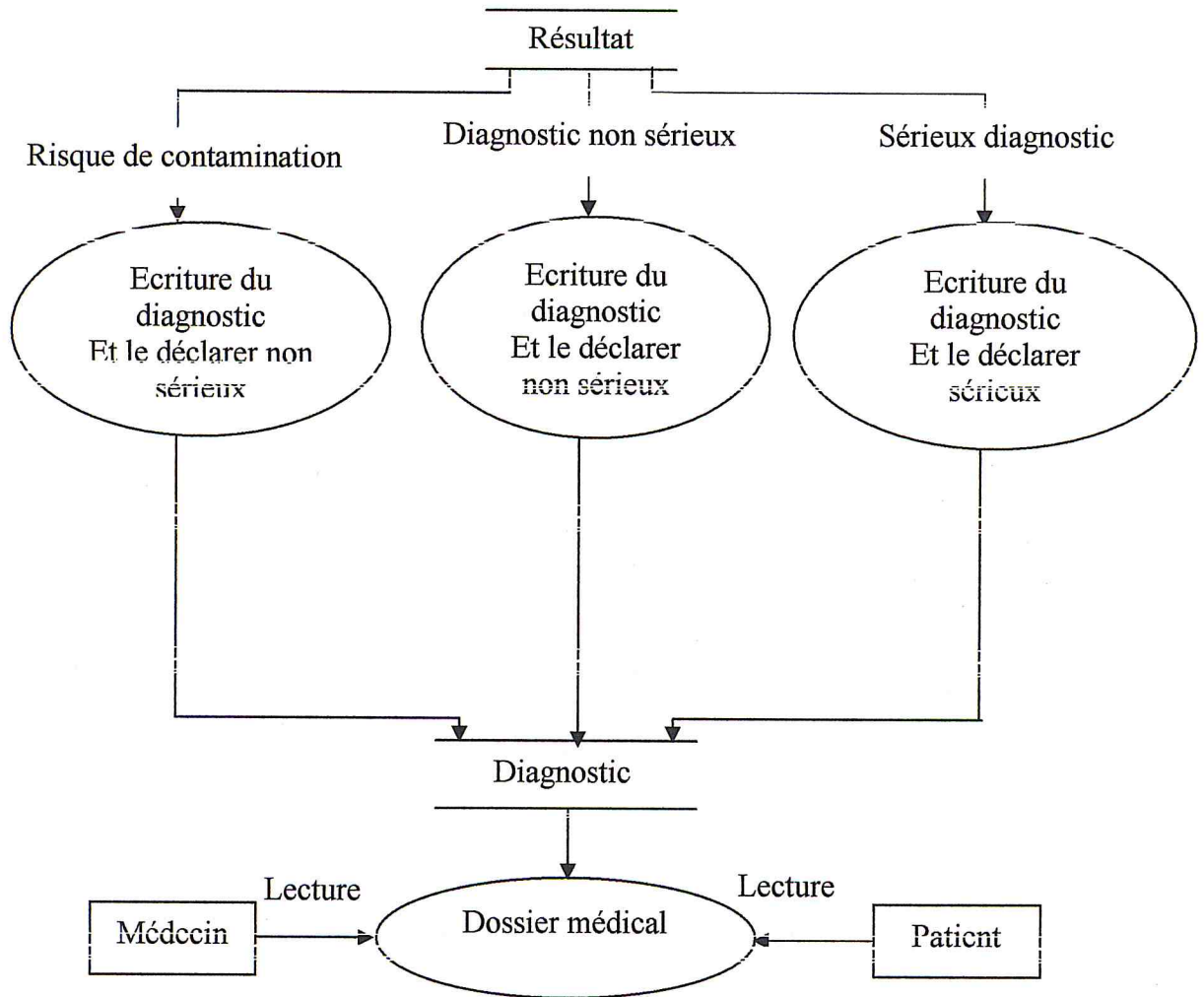
1. Diagramme à flot de données de l'examen du malade :



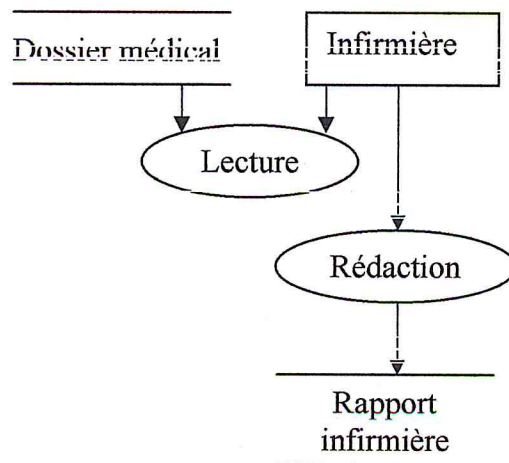
2. Diagramme à flots de données de la création du dossier médicale :



3. Diagramme à flots de données de la mise à jour du dossier médicale :



4. Diagramme de flots de données de la rédaction du rapport infirmière :



## 2. Conception de système :

Le système à construire est un système d'information multi-niveau, son type s'appuie sur des données persistantes stockées dans la base de données (ensemble de tables de modèle relationnel).

### 2.1 Règles de passage entre modèle objet et modèle tables relationnelles :

Représentation des classes d'objet en tables :

-chaque classe est représentée par une ou plusieurs tables.

Représentation des associations en tables :

-chaque association « plusieurs à plusieurs » est représentée par une table distincte.

-chaque association « un à plusieurs » est représenté par une table distincte ou peut être enfouie comme clé étrangère dans la table pour l'une ou l'autre des classes. Pour les associations « un à plusieurs » ou « un à un », il n'y a pas de cycle, on dispose de l'option supplémentaire qui consiste à ranger l'association et l'objet liés dans une seule table.

Ayant conscience que cela peut introduire une redondance et violer des formes normales.

-les noms de rôles sont incorporés en tant que partie du nom de l'attribut de la clé étrangère.

-les associations n-aires ( $n > 2$ ) se représentent par des tables distinctes. Ce qui aide par fois à promouvoir une association n-aires en une classe.

-une association qualifiée se représentent en une table distincte avec au moins trois attributs, la clé primaire de chaque classe liée est le qualificatif.

-les agrégations suivent les mêmes règles que les associations.

Représentation de la généralisation de l'héritage simple en table :

-on représente chaque super-classe et chaque sous-classe par une table.

-s'il n'y a pas de tables de super-classe, les attributs sont dupliqués dans chaque table de sous-classe.

-s'il n'y a pas de table de sous-classe, on apportent tous les attributs des sous-classe dans la super-classe.



2.2 Traduction du modèle objet en base de données relationnelles :

En appliquant les règles de passage au modèle objet, on obtient la représentation logique de la base de données qui seront présent comme suit :

Modèle Objet	Modèle de la table		
Classe personne	Table Personne		
	Nom attribut	Nulle ?	type
Mat Pers	Mat Pers	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A
SF	SF	Non	A
Date Nais	Date Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tel	N°Tel	Non	A

Modèle Objet	Modèle de la table		
Classe Soignant	Table Soignant		
	Nom attribut	Nulle ?	type
Mat Soig	Mat Soig	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A
SF	SF	Non	A
Date Nais	Date Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tel	N°Tel	Non	A
Poste	Poste	Non	A
N°Service	N°Service	Non	A
N°SS	N°SS	Non	A

Modèle Objet	Modèle de la table		
Classe Non-Soignant	Table Non-Soignant		
	Nom attribut	Nulle ?	type
Mat N Soig	Mat N Soig	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A

SF	SF	Non	A
Date-Nais	Date-Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tel	N°Tel	Non	A

Modèle Objet	Modèle de la table		
Classe Médecin	Table Médecin		
	Nom attribut	Nulle ?	type
Mat Med	Mat Med	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A
SF	SF	Non	A
Date-Nais	Date-Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tel	N°Tel	Non	A
Poste	Poste	Non	A
N°Service	N°Service	Non	A
N°SS	N°SS	Non	A
Date- Rec	Date- Rec	Non	Date
Grade	Grade	Non	A
Spécialité	Spécialité	Non	A
SA	SA	Non	A

Modèle Objet	Modèle de la table		
Classe Infirmière	Table Infirmière		
	Nom attribut	Nulle ?	type
Mat Inf	Mat Inf	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A
SF	SF	Non	A
Date Nais	Date Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tel	N°Tel	Non	A
Poste	Poste	Non	A
N°Service	N°Service	Non	A
N°SS	N°SS	Non	A
Date- Rec	Date- Rec	Non	Date
Spécialité	Spécialité	Non	A

Modèle Objet Classe patient	Modèle de la table Table patient		
	Nom attribut	Nulle ?	type
Mat Pat	Mat Pat	Non	ID
Mat Fam	Mat Fam	Non	ID
N°Dos	N°Dos	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A
SF	SF	Non	A
Date-Nais	Date-Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tel	N°Tel	Non	A
Date -E	Date -E	Non	Date
Age	Age	Non	A
NSS	NSS	Non	A
Nom-P	Nom-P	Non	A
Nom-M	Nom-M	Non	A
Nationalité	Nationalité	Non	A
Profession	Profession	Non	A
N°Service	N°Service	Non	A
N°chambre	N°chambre	Non	A
N°lit	N°lit	Non	A
Existence	Existence		A

Modèle Objet Classe Secrétaire	Modèle de la table Table Secrétaire		
	Nom attribut	Nulle ?	type
Mat Sec	Mat Sec	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A
SF	SF	Non	A
Date Nais	Date Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tel	N°Tel	Non	A
N°SS	N°SS	Non	A
Date- Rec	Date- Rec	Non	Date



Modèle Objet	Modèle de la table		
Classe Famille	Table Famille		
	Nom attribut	Nulle ?	type
Mat Fam	Mat Fam	Non	ID
Mat Pat	Mat Pat	Non	ID
Nom	Nom	Non	A
Prénom	Prénom	Non	A
Sexe	Sexe	Non	A
SP	SP	Non	A
Date-Nais	Date-Nais	Non	Date
Lieu -Nais	Lieu -Nais	Non	A
Adr	Adr	Non	A
N°Tcl	N°Tcl	Non	A
Lien de parenté	Lien de parenté	Non	A
Nationalité	Nationalité	Non	A
Profession	Profession	Non	A

Modèle Objet	Modèle de la table		
Classe Dossier médical	Table Dossier médical		
	Nom attribut	Nulle ?	type
N°Dos	N°Dos	Non	ID
Mat Pat	Mat Pat	Non	ID
Mat Sec	Mat Sec	Non	ID
Nom-P	Nom-P	Non	A
Prénom-P	Prénom-P	Non	A
Date-Naiss	Date-Naiss	Non	Date
Lieu-Naiss	Lieu-Naiss	Non	A
Groupe sanguin	Groupe sanguin	Non	A
Date-E	Date-E	Non	Date

Modèle Objet	Modèle de la table		
Classe Diagnostic	Table Diagnostic		
	Nom attribut	Nulle ?	type
N°Diag	N°Diag	Non	ID
N°Dos	N°Dos	Non	ID
Mat Med	Mat Med	Non	ID
Nom Maladie	Nom Maladie	Non	A
Date diagnostic	Date diagnostic	Non	A
Nom médecin soignant	Nom médecin signant	Non	Date
Service	Service	Non	A

Analyses et radio	Analyses et radio	Non	A
Description	Description	Non	A
R-Cont	R-Cont	Nom	A
Type	Type	Non	A
Mode d'entrée	Mode d'entrée	Nom	A

Modèle Objet	Modèle de la table		
Classe Ordonnance	Table Ordonnance		
	Nom attribut	Nulle ?	type
N°Ord	N°Ord	Non	ID
N°Dos	N°Dos	Non	ID
Mat Med	Mat Med	Non	ID
Nom-p	Nom-p	Non	A
Prénom-P	Prénom-P	Non	A
N-M-S	N-M-S	Non	A
Sign-M	Sign-M	Non	A
Date ordonnance	Date ordonnance	Non	Date
Medicament	Medicament	Non	A

Modèle Objet	Modèle de la table		
Classe Rapport Infirmière	Table Rapport Infirmière		
	Nom attribut	Nulle ?	type
N°Rap	N°Rap	Non	ID
N°Des	N°Des	Non	ID
Mat nf	Mat Inf	Non	ID
Nom-P	Nom-P	Non	A
Prénom- P	Prénom- P	Non	A
Tension	Tension	Non	A
Poux	Poux	Non	A
Température	Température	Non	A
Dextro	Dextro	Non	A
Apstixc	Apstixc	Non	A
Date du rapport	Date du rapport	Non	Date
Nom infirmière	Nom infirmière	Non	A

Modèle Objet	Modèle de la table		
Classe Fiche d'observation	Table Fiche d'observation		
	Nom attribut	Nulle ?	type
N°Fic	N°Fic	Non	ID
N°Dos	N°Dos	Non	ID
Mat Med	Mat Med	Non	ID
Nom-P	Nom-P	Non	A
Prénom-P	Prénom-P	Non	A
Décision M	Décision M	Non	A
Date de fiche	Date de fiche	Non	Date

Modèle Objet	Modèle de la table		
Classe Donner soins	Table Donner soins		
	Nom attribut	Nulle ?	type
Mat Med	Mat Med	Non	ID
Mat Pat	Mat Pat	Non	ID

Modèle Objet	Modèle de la table		
Classe Faire intervention	Table Faire intervention		
	Nom attribut	Nulle ?	type
Mat Med	Mat Med	Non	ID
Mat Pat	Mat Pat	Non	ID
Contentment	Consentement	Non	A



### 3. Conception des objets :

Chaque schéma conceptuel est représentée par la définition d'un objet au niveau du modèle objet externe, en les raffinant avec l'ajout des méthodes qui sont directement déduite du modèle dynamique ou des traitement du modèle fonctionnel et en respectant les règles de passage énoncés précédemment. La conception des objets du système est définie par les prototypes des classes suivantes:

Classe : Table Médecin	
Attributs	Méthodes
Mat Med Nom Prénom Sexe SF Date-Nais Licu -Nais Adr N°Tel Poste N°Service N°SS Date- Rec Grade Spécialité Situation actuelle	Prescrire médicament Délivrer médicament

Classe : Table Dossier Médical	
Attributs	Méthodes
N°Dos MatPat MatSec Nom-P Prénom-P Date Naiss Lieu-Naiss Groupe sanguin Date-E	Création Mis à jours Existe Modifier

**Classe : Table Diagnostic**

Attributs	Méthodes
N°Diag	Ecrire
N°Dos	Lire
Mat Med	Modifier
Nom Maladie	
Date diagnostic	
Nom -médecin signant	
Service	
Analyses et radio	
Description	
R-Cont	
Type	
Mode d'entrée	

**Classe : Table Ordonnance**

Attributs	Méthodes
N°Ord	Ecrire
N°Dos	Lire
Mat Med	
Nom-p	
Prénom-P	
N M S	
Sign-M	
Date ordonnance	
Medicament	

**Classe : Table Rapport Infirmière**

Attributs	Méthodes
N°Rap	Ecrire
N°Dos	Lire
Mat Ini	Modifier
Nom-P	
Prénom- P	
Tension	
Poux	
Température	

Dextro	
Apstixe	
Date du rapport	
Nom infirmière	

Classe : Table Fiche d'observation	
Attributs	Méthodes
N°Fic	Ecrire
N°Dos	Lire
Mat Med	Modifier
Nom-P	
Prénom-P	
Décision M	
Date de fiche	

Classe : Table Donner soins	
Attributs	Méthodes
Mat Med	
Mat Pat	

Classe : Table Faire intervention	
Attributs	Méthodes
Mat Med	
Mat Pat	
Contentment	



## **Définition des opérations**

### **Opération existe**

Permet de rechercher le dossier médical du patient afin de vérifier son existence.

### **Opération création**

Si le contrôle de l'existence est négatif donc le dossier médical n'existe pas, alors on possède à sa création.

### **Opération Modifier**

Lorsque une information fait l'objet d'un changement dans le dossier médical.

### **Opération Mis à jour**

Confirmer les changements et mettre à jour la base de données

### **Opération prescrire**

Lorsque le médecin conseil un médicament à un patient (dans une ordonnance).

### **Opération délivrer**

Lorsque le médecin donne un médicament à un patient .

**Conclusion**

L'objectif de la modélisation est de comprendre le problème, son domaine d'application et de construire une conception correcte avant de passer à l'implémentation. Les trois étapes de l'étude ont été étudiées, l'étape statique, dynamique et fonctionnelle. Reste à entamer l'implémentation.

# CHAPITRE V

*Implémentation et  
mise en oeuvre*



## Introduction

Tout développement d'un système d'information se concrétise par sa réalisation et sa mise en oeuvre, c'est en effet l'aboutissement de toute étude.

Au cours de notre étude, plusieurs résultats ont été obtenus, notamment le modèle dynamique, le modèle fonctionnel, les tables relationnelles, la conception des objets du système et la solution informatique.

Cette phase tend à produire un outil qui donne la possibilité d'effectuer les traitements automatisés et de mettre en oeuvre l'application à partir des données issues de l'analyse et de la conception.

### 1. Environnement technique de développement

#### 1.1 Présentation des langages de programmations utilisés

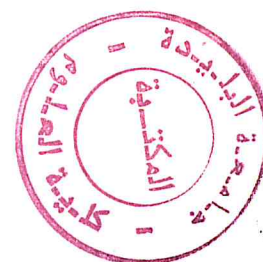
Pour l'implémentation et la gestion de la base de données et le développement du logiciel, on a utilisé les langages suivants :

- Le SGBD ORACLE 8.i
- C++ Builder

### 2. Les différentes applications de logiciel :

Notre logiciel est composé de quatre applications :

- 1 Application d'accès médecin.
- 2 Application d'accès infirmière.
- 3 Application d'accès patient.
- 4 Application de gestion du patient.

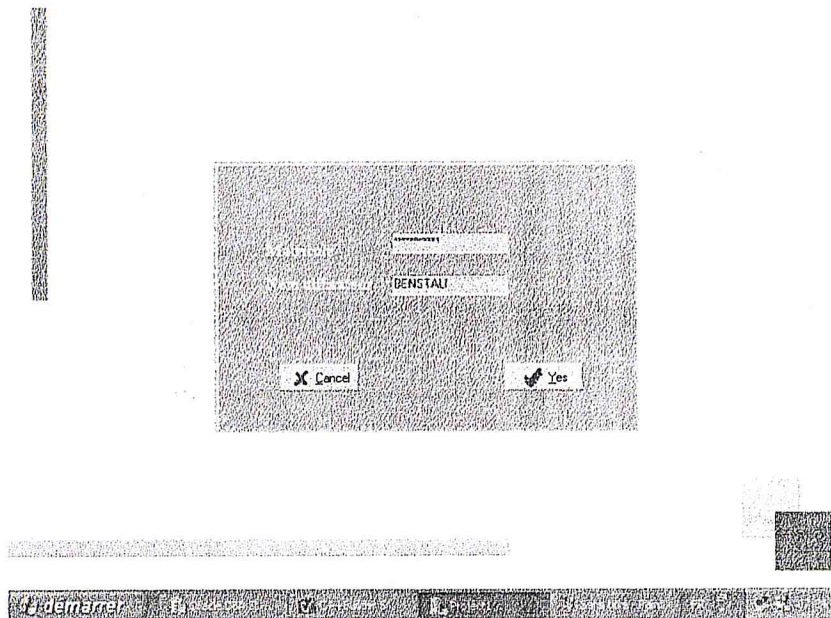




L'accès à chaque application doit être commencer par une phase d'authentification.

### 2.1 La phase d'authentification:

la premiere page est une page ou l'utilisateur doit saisir son matricule et son nom pour l'accès au logiciel.



Après la saisie, un programme s'occupe de vérifier la validation des informations en entrée, en cas d'erreur, un message s'affiche pour l'indiquer.

### Application d'accès médecin:

MMCAR1999003

NOM BENSTALI PRENOM KARIMA SEXE F Situation familiale MARIE

Date Naissance ALGER à 05/04/1955

Adresse 68CITE LES ROSES Numéro de Téléphone 213547896

Date de Recrutement 23/02/1999 N° de Sécurité Sociale 54879632

Poste CHEF SERVICE Spécialité CARDIOLOGIE N° de Service 12 Grade MAITRE ASSISTANT

Situation Actuel ACTIF

A Close

Quitter

A travers cette application, le medecin peut faire les actions suivantes :

- Lire les dossiers medicaux.
- Ecrire les diagnostics.
- Ecrire le leurre en cas d'un diagnostic sérieux.
- Lire les diagnostics.
- Lire le leurre en cas d'un diagnostic sérieux.
- Ecrire des ordonnances.
- Ecrire les fiches d' observations.
- Lire le rapport infirmiere.
- Lire des ordonnances.
- Lire les fiches d' observation.



## Application de gestion des patients:

Form7

Matricule 51999001

Nom BALI Prénom AMINA Sexe F SF MAI

Date Naissance 19/01/1963 Lieu Naissance BLIDA

Numéro Téléphone Adresse 16 RUE DES ROSIER

Date recrutement 19/02/1999 Num Sécurité sociale

LOSE

Form7

A travers cette application la secrétaire peut faire les actions suivantes :

- Création des nouveaux dossiers médicaux .
- La recherche d'un patient.
- Mise à jour des dossiers médicaux.

*Conclusion  
Générale*



## Conclusion générale :

L'objectif de notre étude était de concevoir une base de donnée médical Multi-Vues en appliquant une sécurisation Multi-Niveaux.

La confidentialité des informations médicales est restée longtemps un problème assez important. Grâce au grand pouvoir d'expression du modèle Multi-Vues qui surpasse les autres réalisations, étant donné qu'il permet de considérer qu'un objet est une entité complexe et dont ces composants pouvait avoir différents niveaux de classifications, le problème de confidentialité a été résolu.

Dans le cadre de réalisation de ce projet, et en appliquant le Modèle Multi-Vues on a pu garantir la sécurité des informations médicales et cela :

- En attribuant aux données des niveaux de classification selon le degré de sensibilités de chaque donnée.
- En utilisant une gestion des leurres.
- En accordant aux utilisateurs la possibilité d'accès aux données, après une authentification qui découvrira leur niveau d'habilitation.

Pour la réalisation de la base de donnée qui a été créée sous l'environnement WINDOWS sous le SGBD ORACLE 8 .i, nous avons opté pour la méthode OMT, comme méthode de conception.

Néanmoins, ces résultats devront être placés dans le contexte de ce projet de recherche, qui était marquée par des contraintes de temps externes au projet et par le manque d'expérience et de matériel. Aussi, l'application restera ouverte à tout apports ou suggestions induites par le lecteur de ce document et ils pourront être pris en considération pour les mises à jours ultérieures.

En effet, le projet pourrait bénéficier des extensions liées à l'inférence non autorisée, et à l'utilisation d'une base de connaissance.



*Annexe*

### Règle Patient :

- 1- R patient (p)  $\rightarrow$  P lire Dossier médical (P, P).
- 2- Patient (P)  $\wedge$  Sérieux Diagnostic (P)  $\rightarrow$   $\neg$  Lire diagnostic (P, P).
- 3- R patient (p)  $\wedge$  R Médecin  $\wedge$  Sérieux Diagnostic (P)  $\wedge$   
Appartient Unité Soins (E, O)  $\rightarrow$  Ecrire Diagnostic sérieux.
- 4- R patient (p)  $\wedge$  Sérieux Diagnostic (P)  $\wedge$  Risque de Contamination (P)  $\rightarrow$   
P lire Diagnostic (P, P).

### Règle infirmière :

- 5- Patient (P)  $\wedge$  R infirmière (N)  $\wedge$  Appartient (N, E)  $\wedge$  Appartient (P, O)  $\wedge$   
Unité Soins (E, O)  $\rightarrow$  P lire Dossier médical (N, P).
- 6- Patient (P)  $\wedge$  R infirmière (N)  $\wedge$  Appartient (N, E)  $\wedge$  Appartient (P, O)  $\wedge$   
Unité Soins (E, O)  $\rightarrow$  P écrire Rapport infirmière (N, P).

### Règle Médecin :

- 7- patient (p)  $\wedge$  R Médecin (D)  $\wedge$  Sérieux Diagnostic (P)  $\wedge$  Risque de Contamination (P)  $\wedge$   
Appartient (D, E)  $\wedge$  Appartient (P, O)  $\wedge$  Unité Soins (E, O)  $\rightarrow$  Ecrire Diagnostic Non sérieux.
- 8- patient (p)  $\wedge$  R Médecin (D)  $\wedge$  R Famille (C, P)  $\wedge$  Demande d'examen (C, P)  $\wedge$   
Appartient (D, E)  $\wedge$  Appartient (P, O)  $\wedge$  Unité Soins (E, O)  $\rightarrow$   
P écrire résultat d'examen (D, P).
- 9- patient (p)  $\wedge$  R Médecin  $\wedge$  Appartient (D, E)  $\wedge$  Appartient (P, O)  $\wedge$  Unité Soins (E, O)  
 $\rightarrow$  P Donner soins (D, P).
- 10- patient (p)  $\wedge$  R Médecin  $\wedge$  Appartient (D, E)  $\wedge$  Appartient (P, O)  $\wedge$  Unité Soins (E, O)  
 $\rightarrow$  P lire Dossier médical (P, P).
- 11- patient (p)  $\wedge$  R Médecin  $\wedge$  Appartient (D, E)  $\wedge$  Appartient (P, O)  $\wedge$  Unité Soins (E, O)  
 $\rightarrow$  P Prescrire médicament (D, P).
- 12- patient (p)  $\wedge$  R Médecin  $\wedge$  Exception légale (D)  $\wedge$  Appartient (D, Appartient (P, O)  
 $\wedge$  Unité Soins (E, O)  $\rightarrow$  P délivrer médicament (D, P).
- 13- patient (p)  $\wedge$  R Médecin  $\wedge$  Appartient (D, E)  $\wedge$  Appartient (P, O)  $\wedge$  Unité Soins (E, O)  
 $\rightarrow$   $\neg$  P délivrer médicament (D, P).
- 14- patient (p)  $\wedge$  R Médecin  $\wedge$  Appartient (D, E)  $\wedge$  Appartient (P, O)  $\wedge$  Unité Soins (E, O)  
 $\rightarrow$   $\neg$  P Modifier Rapport infirmière (D, P).
- 15- patient (P)  $\wedge$  R Médecin  $\wedge$  hors état (P)  $\rightarrow$  P faire intervention (D, P).

16- patient (P) ^ R Médecin ^ R Famille ^ hors état (P) ^ consulte (D, F)  
→ P faire intervention (D, P).

17- patient (P) ^ R Médecin ^ hors état (P) ^ cas urgence → P faire intervention (D, P).

18- patient (P) ^ R Médecin ^ Appartient (D, E) ^ Appartient (P, O) ^ Unité Soins (E, O)  
→ P écrire fiche d'observation (D, P).

19- patient (p) ^ R Médecin ^ Appartient (D, E) ^ Appartient (P, O) ^ Unité Soins (E, O)  
→ P écrire ordonnance (D, P).

20- patient (p) ^ R Médecin ^ Appartient (D, E) ^ Appartient (P, O) ^ Unité Soins (E, O)  
→ P signer ordonnance (D, P).



## Bibliographie :

**[BeL75]** :D.Bell et L.Lapadula .Secure Computer Systems : Unified Exposition and Multics Interpretation. Rapport Technique EDS-TR-75-306, MTR-2997, MITRE, Bedford, Mass, 1975.

**[Boa83]**: Air force Studies Board .Multilevel Data Management Security .Rapport Technique, National Research council, Wachington, DC, 1983.

**[BOU 94]** : Bouzegoude Mokrane, George Gardarin & Patrick Valdriez Object :  
Concept Langages- Base de Données –Méthodes –Interfaces.

**[CGY93 a]**: N .and F. Cuppens, A. Gabillon, K. Yazdanian. « MultiVieu Model for Multilevel Object - Oriented Data Bases». of the Ninth Annual Computer Security Application Conference(ACSAC).Orlando , USA 93 .

**[Com90]**: European Economic Community. Information Technology Security Evaluation Criteria (ITSEC).Rapport Technique.

**[CuG98]**: F.Cuppens ,A .Gabillon” Design of Multilevel object –oriented Data bases “Pro of the XII International Symposium on Computer Security(ESORICS).Louvain-la-Neuve, Belgique.

**[CuG99]** : F.Cuppens, A .Gabillon “Logiciel Foundation of Multilevel Databases”. Data and Knowledge Engineering., vol.29, pp.259-291.

**[Cup97]** : F.Cuppens .Conception d’applications Sécurisées .Dans 5<sup>ème</sup> école SIC Réutilisation Interopérabilité et Sécurité, Goiy on Gosas, France ,1997. Campus Thomson.

**[Cup00]** : F.Cuppens .Conception d'applications Sécurisées .Dans 5<sup>ème</sup> école SIC Réutilisation Interopérabilité et Sécurité, Goiy on Gosas, France ,2000. Campus Thomson.

**[Def83]**: Department of Defence. Trusted Computer Systems evaluation Criteria. Rapport Technique, CSC-STD-001-83, 1983.

**[DLS87]**: D.Denning, T. Lunt, R.Shell, M.Heckman, and WShockley. A Multilevel Relational Data Model. In IEEE Symposium on Security and Prevarcy, Oakland, 1997.

**[Gal92]**: T.D.Garvey, T.F. Lunt. Cover Stories for Data Base Security. In S. JaJodia and C. landwehr, editors, Data Base Security, 5: Status and Prospects .North -Holland, 1992 .Results of the IFIP WG11.3 Work shop on Data Base Security.

**[Gro76]**: M.J. Grohn. A model of a Protected Data Management System. Rapport technique ESD -TR-76-289, I .P. Sharp Associates Ltd, Bedford, Mass, 1976.

**[JaK90]** : S. JaJodia and B. Kogan .Integrating an Object -Oriented Data Model with Multi-Level Security .In IEEE Symposium on Security and Privacy,Okland,1986.

**[KTT89]**: T .Keefe, W .Tsai and B. Thuraisingham. SODA: A Secure Object - Oriented Data Base System .Computer and Security, 8(6), 1989.

**[MFW 97]** : Michaele BLAHA, Frédéric EDDY, William PREMERLANI  
« Modélisation et conception orienté objet »MASSON 1997.

**[SMK95]:** M. Schafer, P. Martel, T. Kanawan, and V. Lyons. Multilevel Data Model for the Trusted ONTOS Prototype. In Ninth Annual IFIP WG 11.3 Working conference on Databases Security , Rensselaer Ville ,USA,1995.

**[TFCO93]:** B. Thuraisingham, W. ford, M. Collins, and J. O'Keefe. Design and implementation of a database inference controller. Data and Knowledge Engineering, 11(3), December 1993.