

Université Saâd DAHLAB de Blida



Faculté des Sciences

Département de : Informatique

**Mémoire Présenté par :**

Hamizi Soumia & Nehab Wissam.

En vue d'obtenir le diplôme de Master

**Spécialité : Ingénierie du logiciel**

**Sujet :**



**Conception et développement d'une application web  
de Help Desk pour la gestion d'incidents informatique**

**Soutenu le : 08/07/2010, devant le jury composé de :**

- |                                    |            |
|------------------------------------|------------|
| M <sup>r</sup> . Hadj Yahia        | Président  |
| M <sup>me</sup> . Benstiti.S       | Rapporteur |
| M <sup>me</sup> . Toubaline, Sebti | Examineurs |

MA-004-12-1

# Remerciements

*Avant tout, nous remercions ALLAH qui nous a donné la force  
d'amener ce projet à terme.*

*Nos sincères remerciements s'adressent aussi à nos parents respectifs,  
qui nous ont toujours épaulées, et c'est grâce à leur soutien que nous  
présentons ce projet de fin d'étude.*

*A notre promotrice et chère professeur Mme Benstiti pour son attention  
et son déploiement en notre faveur.*

*Aux membres du jury pour les suggestions et les conseils qu'ils vont  
apporter.*

*Nos remerciements les plus vifs s'adresse à notre encadreur M. Hamadi  
ingénieur informaticien et cadre supérieur à la IRC/ Sonatrach pour  
nous avoir accueillies en tant que stagiaires.*

*A tous ceux dont les acronymes n'apparaissent pas sur cette page, et à  
qui nous témoignons notre profonde gratitude.*



## Résumé

Le Help Desk est une application web conçu pour répondre aux besoins des utilisateurs en termes d'assistance et de maintenance, Il permet d'optimiser le suivi des demandes d'aide, de faciliter le partage et la recherche d'informations, d'assurer la gestion des ressources et de maîtriser le temps de résolution.

Le principal objectif des solutions Help Desk est de limiter les coûts de maintenance et de garantir un service rapide et performant aux utilisateurs.

**Mots clés :** Help Desk, architecture 3-tiers, application web, incident informatique, équipement informatique, UML, Oracle.

## Summary

The Help Desk is a web application designed to reply the needs of users in terms of support and maintenance. It allows to optimize the monitoring of request help, and facilitate sharing and research of information, ensure the management of resources and control the time resolution.

The purpose of Help Desk solutions is to limit maintenance costs and ensure a fast and perform service to users.

**Keywords:** Help Desk, 3-tier architecture, web application, informatic incident, informatic equipment, UML, Oracle.

## ملخص

مكتب المساعدة (Help Desk) هو تطبيق ويب صمم للرد على طلبات المستخدمين من حيث الدعم والصيانة فهو يسمح بتحسين مراقبة طلبات المساعدة، تسهيل عملية التبادل و البحث عن المعلومات، تنظيم و تسيير الموارد.

من بين أهداف مكتب المساعدة (Help Desk) : الحد من تكاليف الصيانة و ضمان خدمة سريعة و فعالة للمستخدمين.

## المصطلحات الأساسية

مكتب المساعدة (Help Desk) ، هندسة 3 مستويات، تطبيق ويب، حادث كمبيوتر، معدات، UML،

Oracle .

# Table des matières

Introduction .....	1
--------------------	---

## **I. Etude préliminaire**

1. Mode de travail usuel .....	4
2. Critiques de la qualité du service .....	5
3. Solution proposée .....	6
4. Objectifs du Help Desk .....	6

## **II. Architecture 3 tiers**

1. Définition .....	10
2. Les principes de l'architecture 3 tiers .....	10
3. Les couches de l'architecture 3 tiers .....	12
4. Les avantages et les limites .....	14

## **III. Modélisation**

1. RUP (Rational Unified Process) .....	18
2. Qu'est ce qu'UML ? .....	17
3. Expression des besoins .....	19
3.1. Diagramme de cas d'utilisation .....	19
4. Analyse .....	25
4.1. Diagramme de séquence .....	25
5. Conception .....	40
5.1. Diagramme de classes .....	40
5.2. Diagramme d'états-transition .....	41



## ***IV. L'implémentation***

1. Les outils d'implémentation .....	45
1.1. SQL*Plus : une interface conviviale .....	45
1.2. PL/SQL : un langage procédural .....	46
1.3. Oracle Application Server .....	46
1.4. Forms Builder : une interface frontale .....	47
1.5. Oracle Reports : un générateur d'état .....	47
2. Réalisation et mise en œuvre de l'application .....	49
2.1. Fenêtre d'authentification .....	49
2.2. L'ajout d'une nouvelle demande d'intervention .....	50
2.3. L'assignation d'un technicien .....	50
2.4. Traitement des demandes .....	51
2.5. Création d'un compte .....	51
2.6. MAJ le profil .....	52
2.7. L'ajout d'un équipement .....	52
2.8. Support technique .....	53
<b>Conclusion générale et perspectives .....</b>	<b>55</b>

## ***V. L'annexe***

<b>Annexe A</b> : contexte de travail .....	<b>I</b>
<b>Annexe B</b> : RUP (Rational Unified Process) .....	<b>IV</b>
<b>Annexe C</b> : UML (Unified Modeling Language) .....	<b>XI</b>
<b>Annexe D</b> : Oracle .....	<b>XIX</b>
<b>Bibliographie .....</b>	<b>56</b>

# Liste des figures

Figure	Page
Cheminement du travail usuel	4
Répartition des couches applicatives dans une architecture trois tiers	11
Les niveaux de l'architecture 3 tiers	13
Diagramme du cas d'utilisation global : gestion d'incidents	20
Diagramme de cas d'utilisation détaillé : demander intervention	21
Diagrammes de cas d'utilisation détaillé : créer un compte	22
Diagramme de cas d'utilisation détaillé : consulter les comptes	22
Diagrammes de cas d'utilisation détaillé : MAJ inventaire informatique	23
Diagramme de cas d'utilisation détaillé : MAJ le support technique	24
Diagramme de séquence : « authentification »	25
Diagramme de séquence : « demander une intervention»	26
Diagramme de séquence : « assigner un technicien»	27
Diagramme de séquence : « traiter la demande»	28
Diagramme de séquence : « consulter les demandes»	29
Diagramme de séquence : « créer un compte »	30
Diagramme de séquence : « modifier le profil	31
Diagramme de séquence : « consulter les comptes »	32
Diagramme de séquence : « ajouter un cas dans le support technique »	33

Diagramme de séquence : « modifier le support technique »	34
Diagramme de séquence : « supprimer un cas du support technique»	35
Diagramme de séquence : « rechercher dans le support technique»	36
Diagramme de séquence : « ajouter un équipement »	37
Diagramme de séquence : « consulter l'inventaire informatique»	38
Diagramme de séquence : « consulter les rapports des statistiques»	39
Diagramme de classes	40
Diagramme d'états-transition	42
Architecture 3 tiers	48
Fenêtre « authentification »	49
Fenêtre « ajouter une nouvelle demande d'intervention »	50
Fenêtre « assigner un technicien »	50
Fenêtre « traiter demandes »	51
Fenêtre « créer un compte »	51
Fenêtre « MAJ le profil »	52
Fenêtre « ajouter un équipement »	52
Fenêtre « support technique »	53



## *Introduction*

Dans une société toujours plus exigeante en termes de qualité et <sup>de</sup>réactivité, les responsables des services informatiques ont aujourd'hui le souci d'apporter la meilleure qualité de service à leurs employés, et aux utilisateurs du système d'information de l'entreprise, il s'agit de répondre à leurs attentes en termes de disponibilité et d'efficacité.

Lorsque l'usage des ordinateurs s'est répandu dans les différents services de l'entreprise, le recours à une aide pour les employés non-informaticiens est devenu indispensable.

De ce fait, le service informatique doit assurer une aide aux utilisateurs des ordinateurs, en intervenant directement auprès d'eux ; mais face à l'émergence et au développement des équipements informatique, une solution informatique doit être développée pour ce service de support technique.

Par conséquent, les entreprises doivent disposer d'un outil fiable et performant qui est "le Help Desk", leur permettant d'automatiser plusieurs tâches, et d'offrir une meilleure qualité de service.

En quelques mots, le support technique (bureau d'aide ou Help desk en anglais) ; est une structure d'informations créée pour répondre à la croissance des technologies de l'information; c'est un système de services proposés à la demande des utilisateurs, qui répond aux exigences opérationnelles et s'adapte à l'évolution des besoins de l'entreprise, c'est le cas de notre organisme d'accueil :

L'entreprise Sonatrach souhaite se doter d'une application web "Help Desk", pour la prise en charge des demandes d'interventions adressées au département informatique et pour la gestion des équipements informatiques. C'est sur ce point que nous allons intervenir afin d'apporter une solution optimale pour la gestion des incidents informatique.

**Structuration du mémoire** : la démarche de réalisation de notre projet est répartie dans les quatre chapitres suivants:

### **Chapitre 1 : *L'étude préliminaire***

Cette partie démontre les points négatifs du mode de travail usuel, et cela en décrivant les interactions entre les acteurs du système, afin de cerner la problématique et les préjudices qu'elle apporte au service informatique et à toute l'entreprise, et d'adopter la meilleure solution.

### **Chapitre 2 : *l'architecture 3 tiers***

Ce chapitre décrit les caractéristiques de l'architecture 3 tiers ; qui est une architecture fondamentale pour toute application web de Help Desk.

### **Chapitre 3 : *la modélisation***

Cette phase détaille les caractéristiques de notre application via le langage de modélisation UML, suivant la démarche RUP, pour présenter l'architecture et le fonctionnement générale du logiciel.

Pour réaliser la modélisation, quatre diagrammes seront utilisés :

- Diagramme de cas d'utilisation : il définit les grandes fonctionnalités réalisées par les différents acteurs du système.
- Diagramme de classes : c'est une représentation détaillée des classes existantes, sur lesquelles repose la phase d'implémentation.
- Diagramme de séquences : illustre tous les scénarios possibles du système.
- Diagrammes d'états-transition : représente les différents états que peut avoir un objet.

### **Chapitre 4 : *l'implémentation***

C'est la phase finale du développement, elle décrit les outils utilisés pour l'implémentation, et les fonctionnalités générales de notre application Help desk à travers ses différentes fenêtres.

# *I. Etude préliminaire*

---

## **L'objectif du chapitre**

Permet d'étudier le processus actuel de la gestion des incidents informatique, et d'enregistrer les besoins réels de l'entreprise de façon claire.

## **Dans ce chapitre :**

- Mode de travail usuel.
- Critiques de la qualité de service.
- Solution proposée.
- Objectifs du Help Desk.

## **Eléments mis en jeu :**

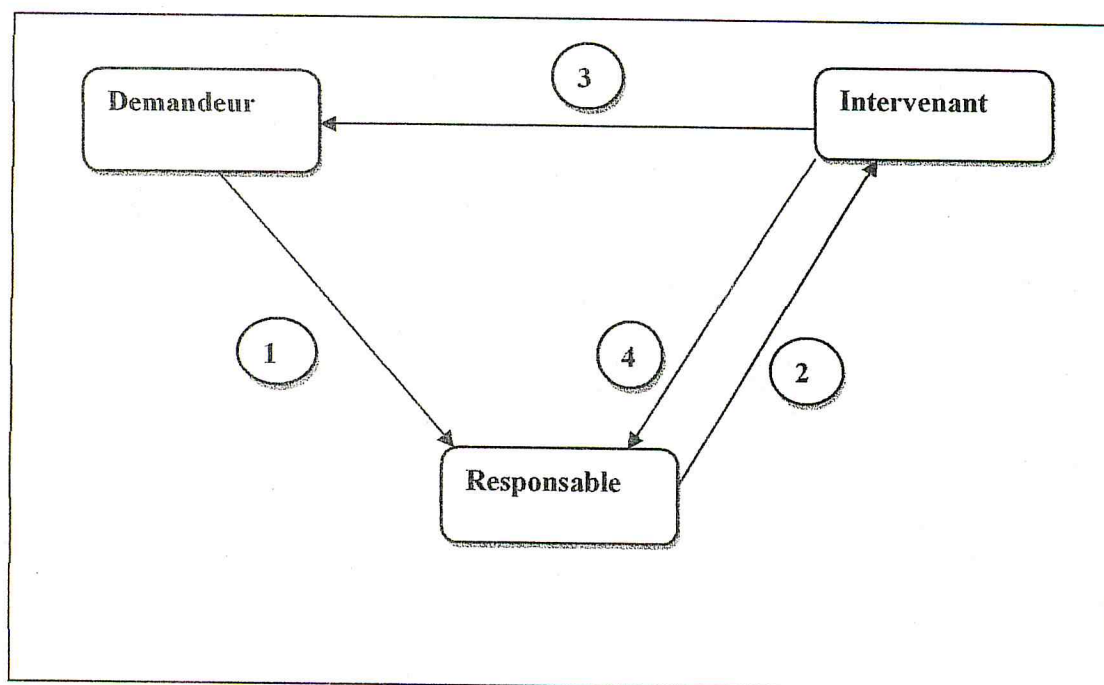
- ✓ Acteur.
- ✓ Incident informatique.
- ✓ Help Desk.



L'étude effectuée au sein du Service support Technique, du département informatique (voir annexe A), nous a permis d'observer le processus usuel du travail et de représenter les différentes interactions entre les acteurs finaux du système, afin d'enregistrer toute problématique rencontrée lors de sa réalisation, et proposer ainsi une meilleure solution pour satisfaire les objectifs de l'entreprise.

### 1. Mode de travail usuel :

Le schéma ci-dessous résume le cheminement du travail accompli pour la gestion des incidents, et les flux échangés entre les trois principaux acteurs du système :



*Figure 1 : Cheminement du travail usuel*

**Le responsable** : Chef du département informatique, reçoit les demandes et les assigne aux techniciens.

**Le demandeur** : Demande la résolution d'un problème informatique.

**L'intervenant** : Technicien qui intervient pour résoudre un problème.

N°	Flux	Description
1	Demande d'intervention	Le demandeur remplit une fiche technique comportant le nom, la structure, la description du problème, date/heure de la demande... etc.
2	Assignment d'un intervenant	Le responsable cherche dans son équipe technique, un technicien "disponible", et dont les compétences s'adaptent le mieux avec la nature du problème posé.
3	L'intervention	Le technicien désigné se déplace à la structure concernée pour intervenir auprès du demandeur, équipé, si nécessaire, de ses outils (CD Windows, câble réseau, antivirus... etc.)
4	Diagnostic d'intervention	Le technicien décrit au responsable du département informatique le résultat de son intervention en déterminant:  Le type d'intervention, le processus de résolution, l'heure du début et de la fin d'intervention... etc.  ✓ cette description est utilisée par le responsable du département informatique pour suivre l'état des interventions.

***Tableau 1 : flux d'informations***

## **2. Critiques de la qualité du service actuel :**

L'étude du cheminement du travail usuel pour la gestion des incidents informatique (figure 1) nous a permis de comprendre la situation actuelle et d'enregistrer ses différentes failles:

- ✓ **Une procédure manuelle** : la description du problème se fait en remplissant une fiche technique, et comme le nombre de demandeurs est trop élevé, cela entraîne un risque de désorganisation dans la gestion des incidents ainsi que la non prise en charge de toutes les demandes d'intervention.
- ✓ **L'indisponibilité** : le nombre de techniciens est réduit par rapport au nombre des demandes (en moyenne le département reçoit 30 demandes par jour, pour une équipe de 5 techniciens).
- ✓ **Le déplacement** : pour chaque demande d'intervention, l'utilisateur doit se rapprocher auprès du département informatique, et pour chaque intervention, le technicien désigné doit aussi se rapprocher auprès de la structure concernée, ce qui implique une perte de temps considérable.
- ✓ **Perte de temps** dans le traitement des problèmes déjà résolus de la même sorte.

Tous ces points démontrent le besoin de recourir à un système informatique pour couvrir tout le processus de gestion des incidents informatique.

### **3. Solution proposée :**

Après concertation avec le responsable du département informatique, nous avons opté pour le développement d'une solution logicielle avec la réalisation d'un Help Desk Informatique, c'est une application web qui permet la prise en charge et le suivi de toutes les demandes d'intervention, dans le but de les satisfaire, en offrant des modules et des options qui garantissent une gestion optimale.

### **4. Objectifs du Helpdesk:**

Le principal objectif du Help Desk est de fournir un service fiable et efficace aux utilisateurs, en améliorant considérablement la qualité de service existant et en permettant une grande transparence opérationnelle.



#### 4.1. Fonctionnalités du Help Desk:

Les fonctions du Help Desk sont diverses :

- ✓ *La centralisation des demandes* : Centraliser et enregistrer toutes les demandes en provenance des utilisateurs vers l'équipe informatique.
- ✓ *La gestion immédiate des problèmes* : Le traitement des problèmes est la fonction primordiale du processus, qui consiste à trouver l'origine d'un incident, et de présenter la solution proposée.
- ✓ *L'assignation des demandes* : facilite l'attribution des demandes d'intervention aux techniciens de l'équipe informatique.
- ✓ *La prise en charge totale* : contribue à assurer la gestion des incidents du début à la fin, avec le traitement et le suivi de toutes les demandes en toute rapidité.
- ✓ *Garder l'historique des incidents* : sauvegarder l'inventaire informatique de l'entreprise, pour garantir une meilleure gestion du parc informatique.
- ✓ *Le Reporting*: l'outil Help Desk, permet de faire des requêtes à partir des données extraites de l'application, de réaliser des analyses multidimensionnelles, de créer des tableaux de bords d'aide à la décision et d'avoir des statistiques et des rapports pertinents (nombre de demandes par technicien, les pannes les plus fréquentes ... etc.), pour une meilleure visibilité du processus de gestion d'incidents informatique.
- ✓ *La Réduction des coûts de maintenance* : facilite le partage et la circulation des informations pour réduire les coûts d'administration et améliorer la qualité du service afin d'accroître la productivité.

En effet, la performance de notre application Help Desk, s'appuie sur les fonctionnalités citées ci-dessus, sans épargner son aspect ergonomique, grâce à l'attractivité et la convivialité des interfaces utilisateurs qui favorisent l'autonomie du travail.

*Cette partie nous a permis d'avoir une vision analytique sur le processus de gestion des incidents informatique dans le Service Support Technique, et de proposer par la suite une solution logicielle qui tiendra compte des difficultés enregistrées dans le mode de travail usuel.*

## ***II. L'architecture 3 tiers***

---

### **L'objectif du chapitre**

Donne une description approfondie de l'architecture 3 tiers, tout en illustrant son impact sur le développement de notre application Help Desk.

### **Dans ce chapitre :**

- Principes de l'architecture 3 tiers.
- Couches de l'architecture 3 tiers.
- Avantages et limites.

### **Éléments mis en jeu :**

- ✓ Couche
- ✓ Client, serveur BD, serveur web.



*Tout système informatique est fondé sur une architecture bien définie, et notre solution logicielle sera conçue selon une architecture 3 tiers, qui repose sur les possibilités offertes par le développement des réseaux et de leur communication et sur le principe de la séparation des données, des programmes et des interfaces hommes-machines servant à la présentation de l'information*

**1. Définition :** L'architecture 3 tiers, également appelée architecture client-serveur de deuxième génération ou client-serveur distribué, sépare l'application en 3 niveaux de services distincts.

Plusieurs postes clients connectés à un réseau Intranet ou internet, ces postes dialoguent tous avec un serveur Web qui lui-même est en communication avec un serveur de base de données. Ce dernier héberge l'information dont les utilisateurs ont besoins pour un traitement particulier. [Arch1]

## **2. Les principes de l'architecture 3 tiers :**

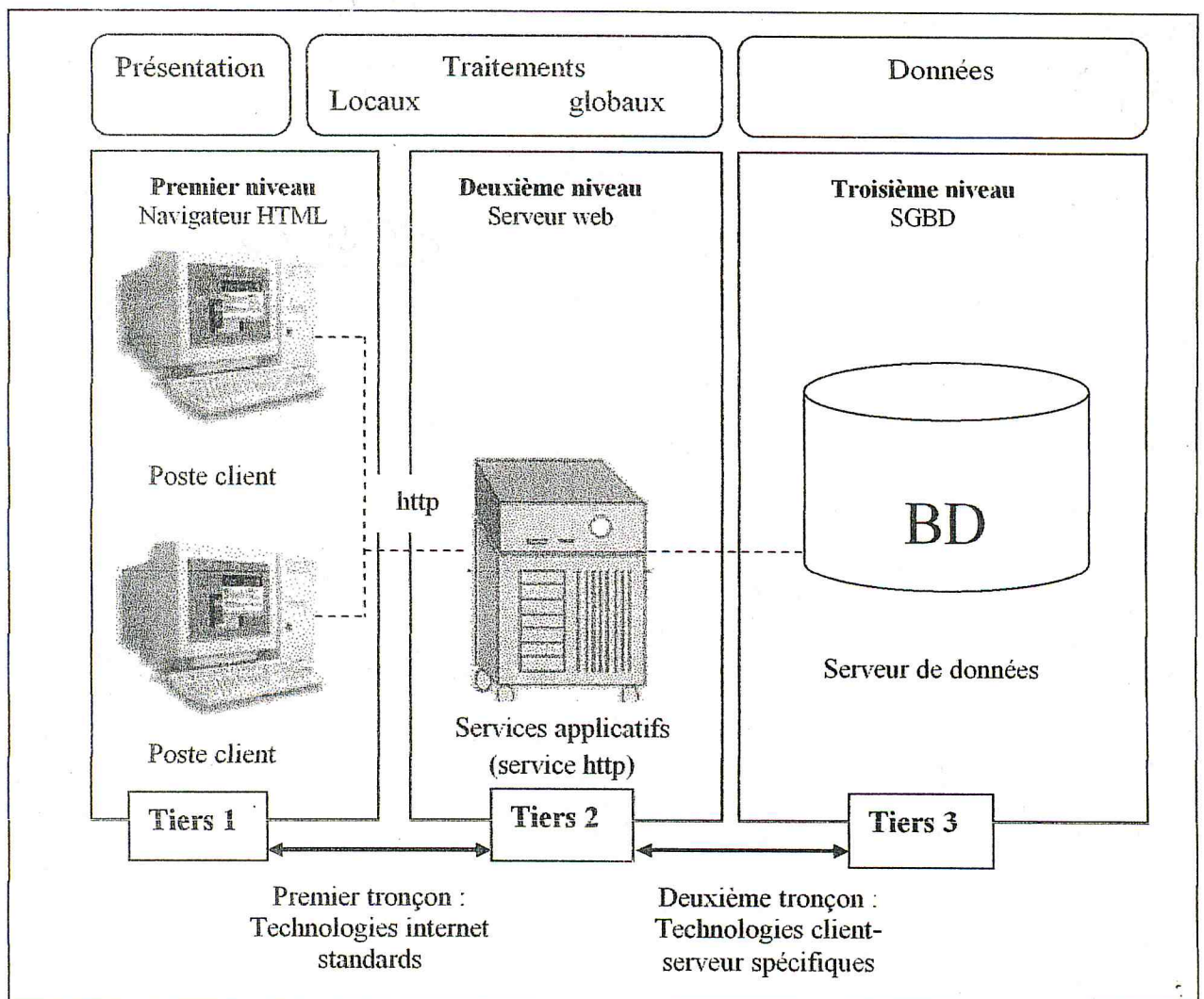
L'architecture 3 tiers applique les principes suivants :

- Les données sont gérées de façon centralisées
- La présentation est toujours prise en charge par le poste client
- La logique applicative est prise en charge par un serveur intermédiaire
- 

Le poste client prend la forme d'un simple navigateur Web, le service applicatif est assuré par un serveur HTTP et la communication avec le SGBD met en œuvre les mécanismes bien connus des applications client-serveur de la première génération.

Ce type d'architecture fait une distinction nette entre deux tronçons de communication indépendants et délimités par le serveur HTTP :

- le premier tronçon relie le poste client au serveur Web pour permettre l'interaction avec l'utilisateur et la visualisation des résultats. On l'appelle **circuit froid** et n'est composé que de standards (principalement HTML et HTTP). Le serveur Web tient le rôle de "façade HTTP".
- le deuxième tronçon permet la collecte des données, il est aussi appelé **circuit chaud**. Les mécanismes utilisés sont comparables à ceux mis en œuvre pour une application deux tiers. Ils ne franchissent jamais la façade HTTP et, de ce fait, peuvent évoluer sans impacter la configuration des postes clients.



*Figure 2 : Répartition des couches applicatives dans une architecture trois tiers*

La figure-2 montre l'organisation d'une telle architecture trois tiers :

**Tiers 1 :** Possède essentiellement des fonctions d'affichage, effectue des requêtes vers le serveur. Les résultats de ses requêtes sont ensuite affichés dans un navigateur web.

**Tiers 2 :** C'est un serveur qui effectue le traitement principal ou effectue des requêtes vers le serveur de données.

**Tiers 3:** C'est un serveur de donnée qui contient la base de données de l'application.

L'enchaînement standard de cette architecture suit les étapes suivantes :

1. Le client effectue une action (connexion, requête,... etc.).
2. Le serveur traite cette action en fonction de l'utilisateur (client).
3. Le serveur effectue des requêtes auprès de la base de données.
4. La base de données renvoie les informations demandées.
5. Le serveur met en forme ces informations (dans une page web) pour le client.
6. Le client (le navigateur Web) affiche les informations reçues. [Arch2].

### **3. Les couches de l'architecture 3 tiers :**

L'architecture 3 tiers se compose de trois éléments, ou plutôt de trois couches, chacune liée entre elles de façon logique. Ces trois couches sont les suivantes :

- 3.1. **La couche de données** liée au serveur de base de données (SGBD), elle correspond aux données qui sont destinées à être conservées de manière définitive.

#### **3.1.1. Caractéristiques :**

- ▀ Propose l'accès aux données.
- ▀ Le stockage des données (fichiers...etc.).
- ▀ Fournit au serveur web (La couche fonctionnelle) les données dont il a besoin.
- ▀ Évite la réplique des données pour assurer leur uniformité.



3.2. **La couche fonctionnelle** (ou couche métier ou **middleware**) liée au serveur, qui dans de nombreux cas est un serveur Web muni d'extensions applicatives.

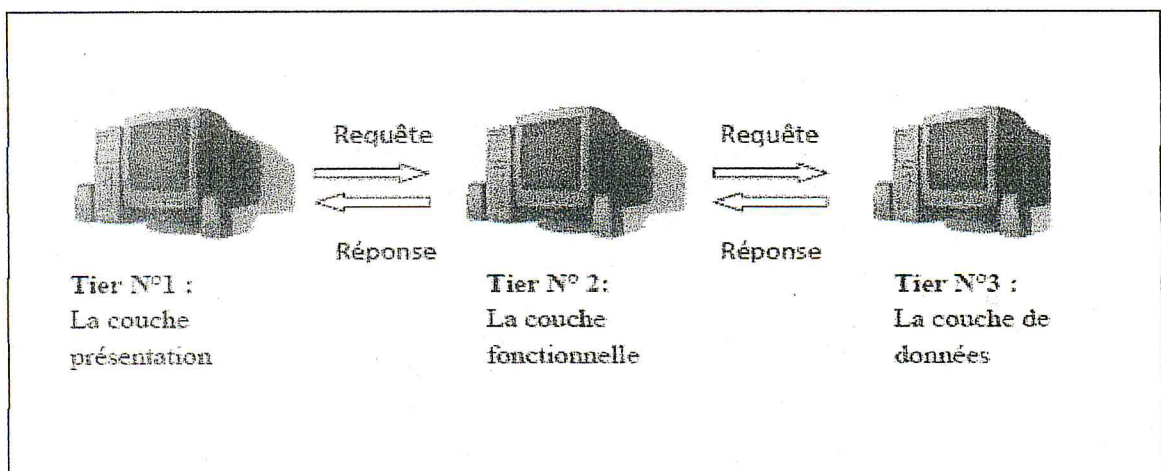
**3.2.1. Caractéristiques :**

- effectue le contrôle de processus métier, où la logique et les règles fonctionnelles sont exécutées.
- Effectue la présentation des données au format HTML.
- fournit la ressource demandée.

3.3. **La couche présentation** (ou cliente) c'est l'ordinateur demandeur de ressources, équipée d'une interface utilisateur chargée de la présentation, associée au client qui de ce fait est dit "léger". Il est souvent constitué d'un simple navigateur Internet.

**3.3.1. Caractéristiques :**

- contient l'interface utilisateur.
- l'affichage des données sur le poste de travail du client.
- le dialogue avec l'utilisateur (interface homme-machine).
- Plusieurs interfaces possibles pour une même application. *[Arch1]*



*Figure 3 : les niveaux l'architecture 3 tiers*

Tous ces niveaux étant indépendants, ils peuvent être implantés sur des machines différentes, alors :

- ✓ Le poste client ne supporte plus l'ensemble des traitements, il est moins sollicité et peut être moins évolué, donc moins coûteux,
- ✓ Les ressources présentes sur le réseau sont mieux exploitées, puisque les traitements applicatifs peuvent être partagés ou regroupés
- ✓ La fiabilité et les performances de certains traitements se trouvent améliorés par leur centralisation

Il est possible et souvent probable que ces deux serveurs soient une seule et même machine, dans ce cas, on retombe sur une architecture *2-tiers*. [Arch2]

#### 4. Les avantages

Les principaux avantages d'une telle architecture sont les suivants :

- ✓ L'installation sur le poste client est plus simple.
- ✓ Une plus grande sécurité
- ✓ La maintenance des traitements est facilitée,
- ✓ Le travail en équipe est optimisé. [Arch2]
- ✓ Les requêtes clients vers le serveur sont d'une plus grande flexibilité; en effet les appels clients ne spécifient que des paramètres et des structures de données.
- ✓ des modifications peuvent être faites au niveau du SGBD sans que cela impacte la couche client.
- ✓ la séparation qui existe entre la *couche présentation*, la *couche fonctionnelle* et la *couche de données* permet une spécialisation des développeurs sur chaque tiers de l'architecture. [Arch1]

#### 5. Limites de l'architecture 3 tiers :

L'architecture trois tiers a corrigé les excès du client lourd en centralisant une grande partie de la logique applicative sur un serveur HTTP. Le poste client, qui ne prend en charge que la présentation et les contrôles de saisie, s'est trouvé ainsi soulagé et plus simple à gérer.

Par contre, le serveur HTTP constitue la pierre angulaire de l'architecture et se trouve souvent fortement sollicité et il est difficile de répartir la charge entre client et serveur. On se retrouve confronté aux épineux problèmes de dimensionnement du serveur et de gestion de la montée en charge rappelant l'époque des mainframes.

De plus, les solutions mises en œuvre sont relativement complexes à maintenir et la gestion des sessions est compliquée.

Les contraintes semblent inversées par rapport à celles rencontrées avec les architectures deux tiers : le client est soulagé, mais le serveur est fortement sollicité.

Le phénomène fait penser à un retour de balancier. *[Arch2]*

*Dans l'architecture trois tiers, les données sont hébergées dans un serveur de base de données dans lequel tourne un ou des SGBD (en général relationnel) , les programmes (applications et/ou services Web) résident dans un serveur web , tandis que les interfaces homme-machine restent au niveau du poste client. Le tout fonctionnant au sein d'un réseau local ou distant.*



# *III. Modélisation*

---

## **L'objectif du chapitre :**

L'objectif de la modélisation consiste à concevoir le futur système et représenter la nouvelle solution logicielle conformément aux besoins exprimés.

## **Dans ce chapitre :**

- RUP (Rational Unified Process).
- Qu'est ce que UML.
- Expression des besoins.
- Analyse.
- Conception.

## **Eléments mis en jeu :**

- ✓ RUP.
- ✓ UML.
- ✓ Acteur
- ✓ Diagramme de cas d'utilisation, diagramme de classes, diagramme de séquence, diagramme d'états-transition.

*La modélisation est une technique d'ingénierie qui permet de représenter un système par l'établissement de modèles pour mettre au point une solution à un problème.*

*Modéliser un système avant sa réalisation permet de représenter son fonctionnement, c'est également un bon moyen de maîtriser sa complexité et d'assurer sa cohérence.*

### **1. RUP (Rational Unified Process) :**

RUP est défini par ses créateurs comme un framework de processus génériques destiné à gérer l'ensemble des aspects d'un projet de développement de logiciel. RUP agrège des bonnes pratiques formalisées qui sont destinées aux différents acteurs d'un projet. Chacun de ces sous-processus s'intègre dans un workflow lui aussi complètement spécifié.

RUP est néanmoins décrit par ses auteurs comme un métaprocessus. C'est-à-dire qu'il est possible d'instancier RUP. En clair, cela signifie que RUP peut être adapté à tous les types de projets, pour tenir compte :

- De la taille de projet.
- Du niveau de compétence initiale des acteurs (par exemple dans le cas où l'entreprise n'utiliserait pas de processus de développement).

RUP est une démarche de développement qui est souvent utilisé conjointement au langage UML [RUP].

### **2. Qu'est qu' UML ?**

UML est une notation graphique conçue pour représenter, spécifier, construire les systèmes logiciels. Ses deux principaux objectifs sont la modélisation de systèmes utilisant les techniques orientées objet, depuis la conception jusqu'à la maintenance, et la création d'un langage abstrait compréhensible par l'homme et interprétable par les machines.

UML s'adapte à tous les domaines d'application et à tous les supports. Il permet de construire plusieurs modèles d'un système, chacun mettant en valeur des aspects différents : fonctionnel, statiques, dynamiques et organisationnels. UML est devenu un langage incontournable dans les projets de développement. [UML2]

### **3. Déroulement du processus :**

Les activités du RUP, sont des étapes dans le développement d'un logiciel, Chaque activité est répétée autant de fois qu'il y a d'itérations.

#### **3.1 Expression des besoins :**

L'activité de recueil et d'expression des besoins débouche sur ce que doit faire le système (question \_ QUOI ? \_) grâce à l'utilisation des cas d'utilisation pour schématiser les besoins.

Les cas d'utilisation sont décomposés en scénarios d'usage du système, dans lesquels l'utilisateur raconte ce qu'il fait grâce au système et ses interactions avec lui. [Proc]

##### **3.1.1 Diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation permet de recueillir, d'analyser et d'organiser les besoins. C'est Avec lui que débute l'étape d'analyse d'un système

##### **Qu'est ce qu'un cas d'utilisation ?**

Un cas d'utilisation est une manière spécifique d'utiliser un système. Les acteurs sont à l'extérieur du système ; il modélise tout ce qui interagit avec le système. Un cas d'utilisation présente un service pour l'acteur qui l'initie. [UML2]

Les diagrammes ci-dessous, montrent les grandes fonctionnalités du processus de travail de Help Desk.

##### **3.1.1.1 Diagramme du cas d'utilisation global « gestion d'incidents »**

**NB :** Tous les cas d'utilisation inclus le cas « authentification »



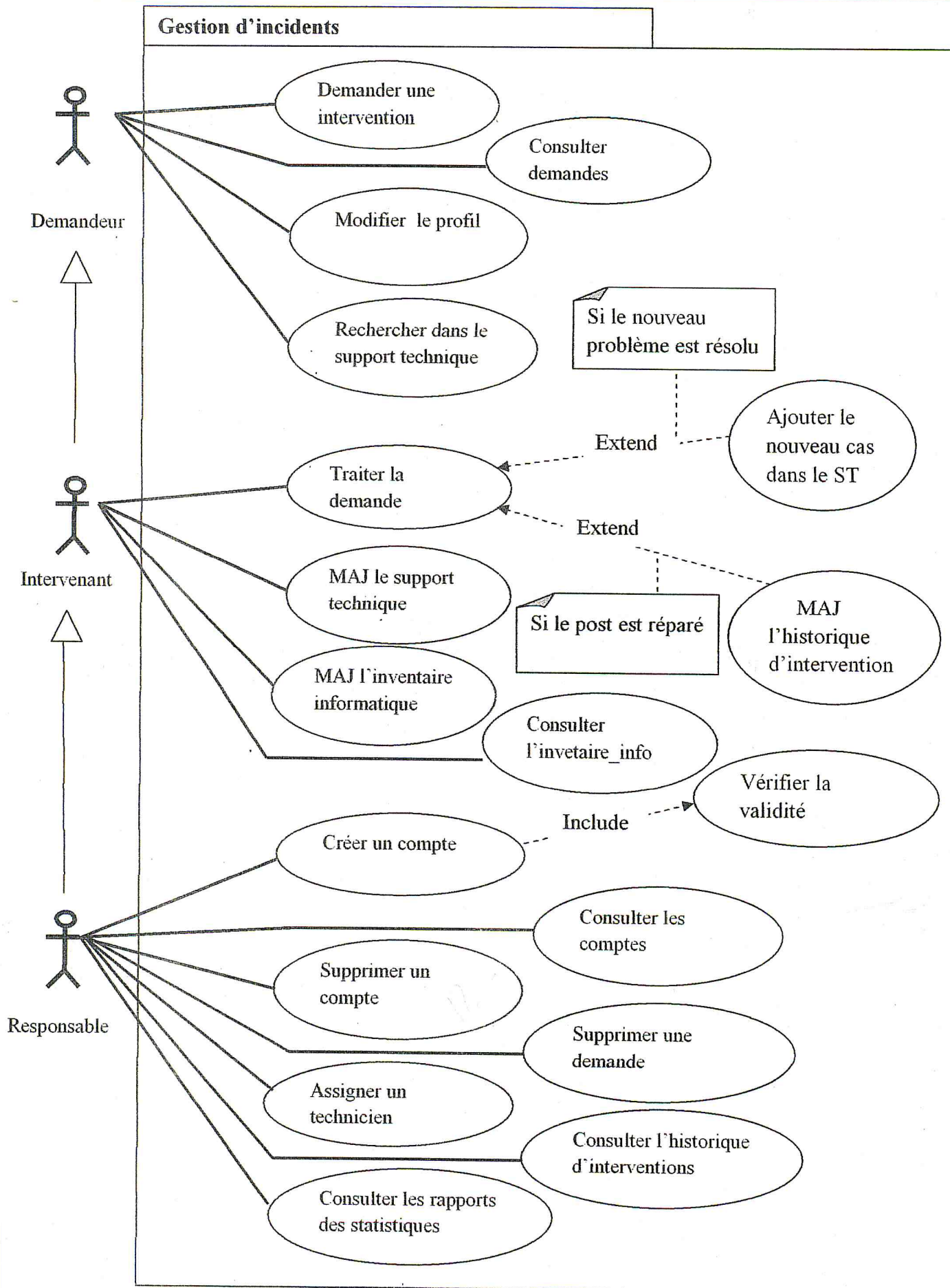
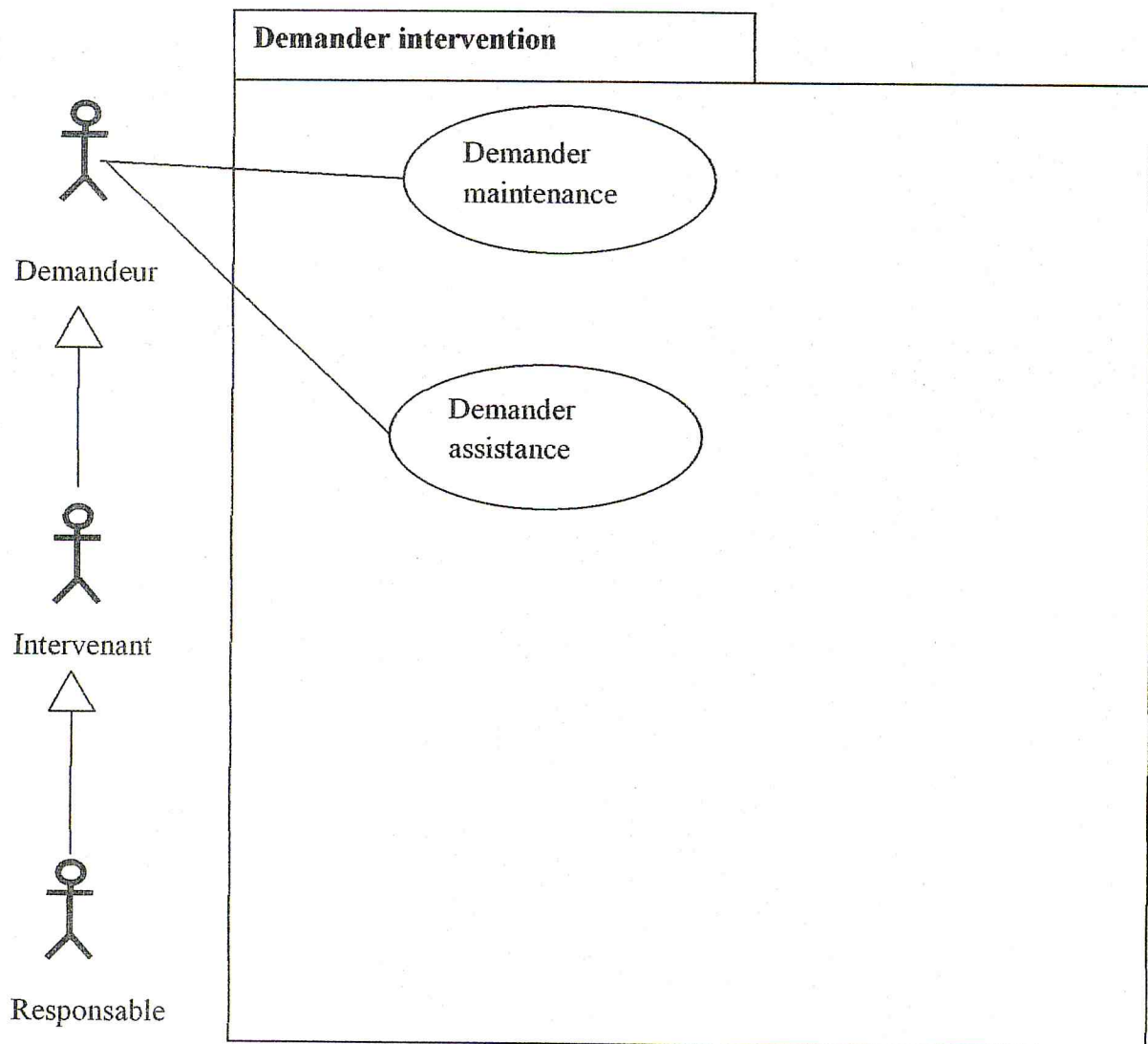


Figure 4 : Diagramme du cas d'utilisation global : gestion d'incidents

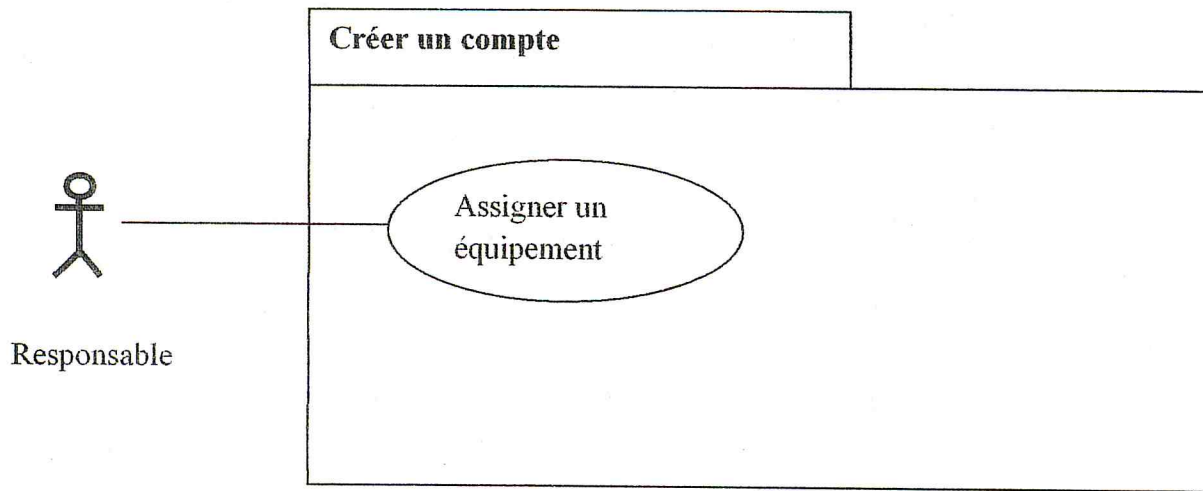


3.1.1.2 Diagramme détaillé du cas d'utilisation «demander intervention»



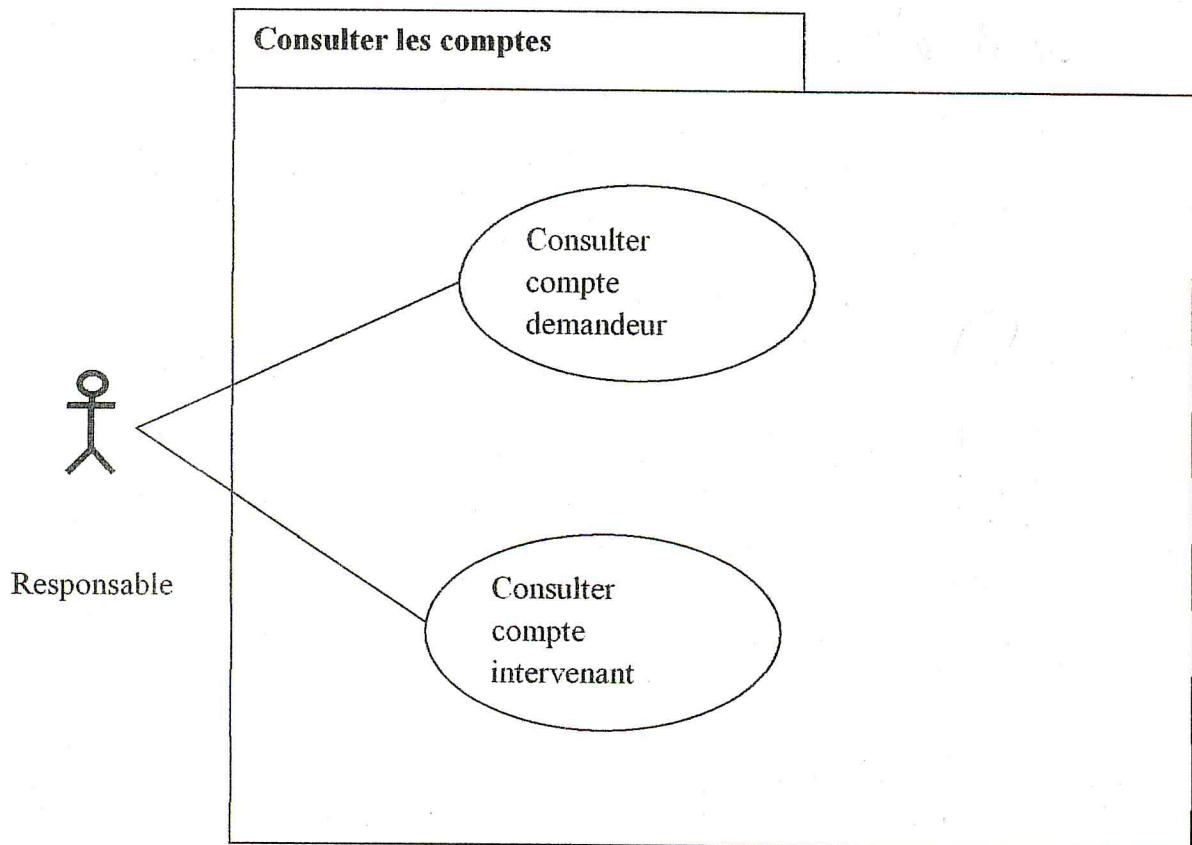
*Figure 5 : Diagramme détaillé du cas d'utilisation: demander intervention*

**3.1.1.3 Diagramme détaillé du cas d'utilisation « créer un compte » :**



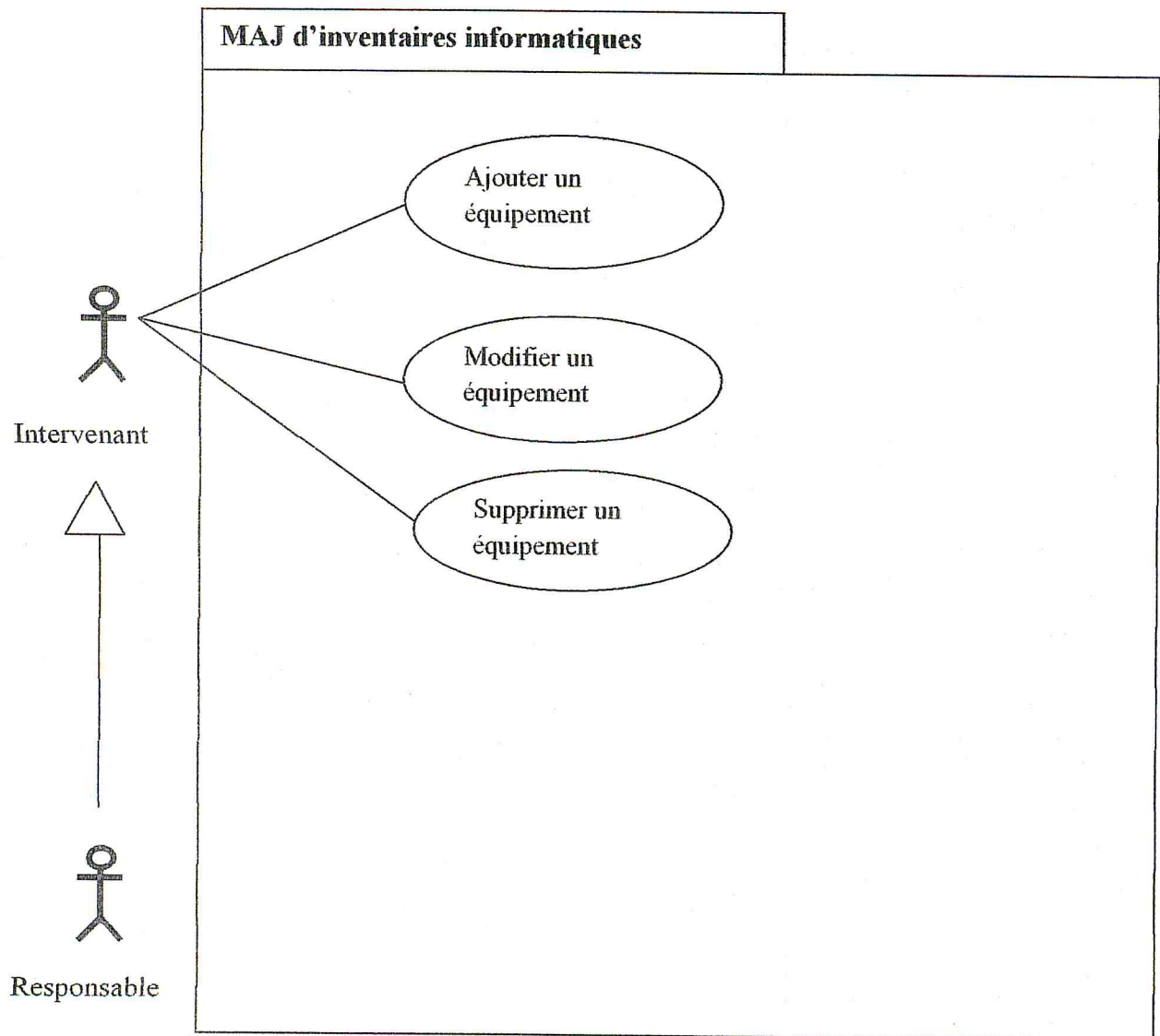
*Figure 6 : Diagramme détaillé du cas d'utilisation: créer un compte*

**3.1.1.4 Diagramme détaillé du cas d'utilisation « consulter les comptes » :**



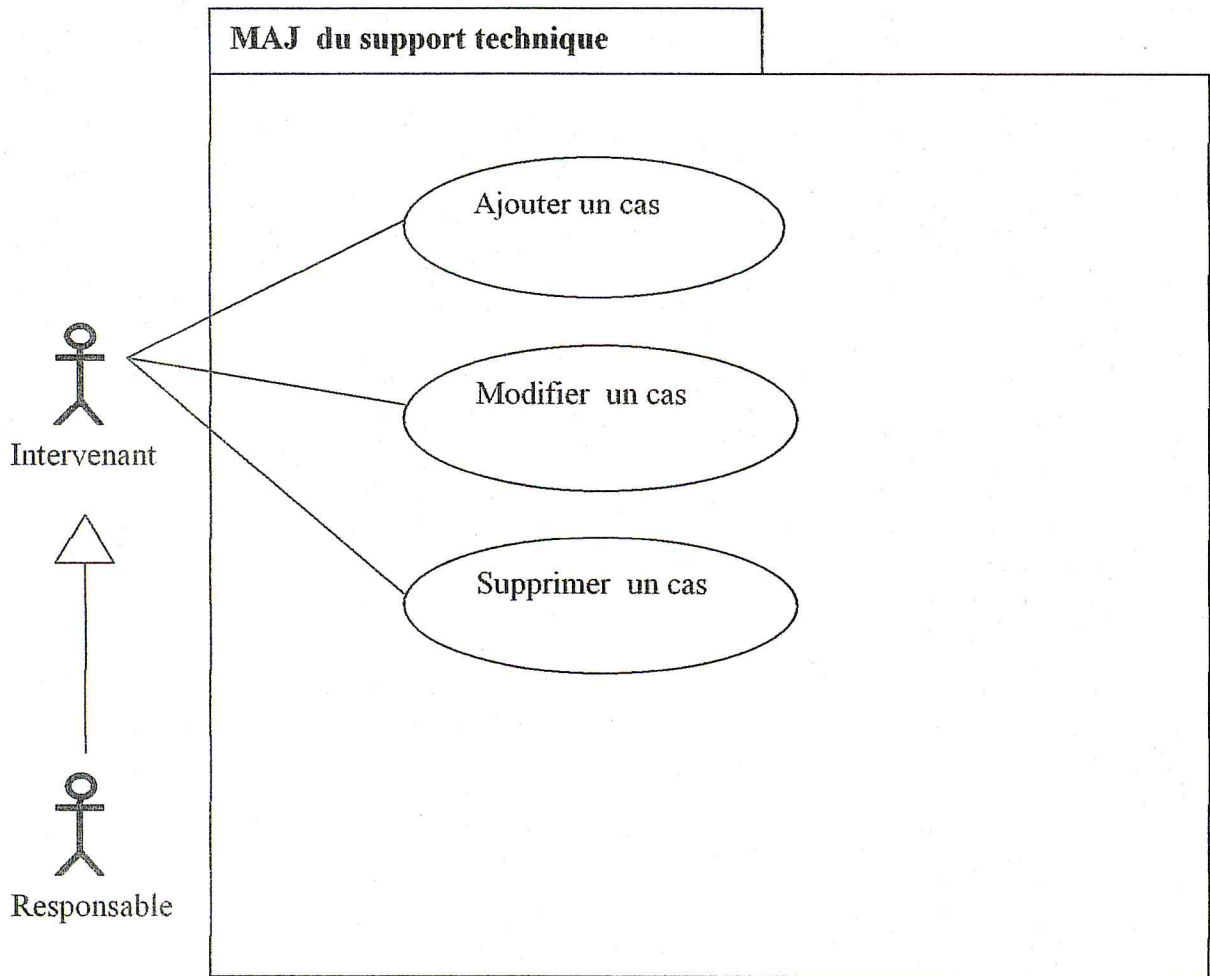
*Figure7 : Diagramme détaillé du cas d'utilisation: consulter les comptes*

3.1.1.5 Diagramme détaillé du cas d'utilisation « MAJ l'inventaire informatique »



*Figure 8 : Diagramme détaillé du cas d'utilisation: MAJ inventaire informatique*

3.1.1.6 Diagramme détaillé du cas d'utilisation « MAJ le support technique »



*Figure 9 : Diagramme détaillé du cas d'utilisation: MAJ le support technique*



### 3.2 Analyse :

L'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, etc. L'analyse répond donc à la question « *que faut-il faire ?* » et a pour but de se doter d'une vision claire et rigoureuse du problème posé et du système à réaliser en déterminant ses éléments et leurs interactions.

L'analyse livre une spécification plus précise des besoins grâce à l'utilisation du diagramme de séquence. Elle peut être envisagée comme une première ébauche du modèle de conception. *[Proc]*

#### 3.2.1 Diagramme de séquence

Le diagramme de séquence montre les interactions sous un angle temporel, et plus particulièrement le séquençage temporel de messages échangés entre des lignes de vie. *[UML2]*

##### 3.2.1.1 Diagramme de séquence du cas d'utilisation « authentication »

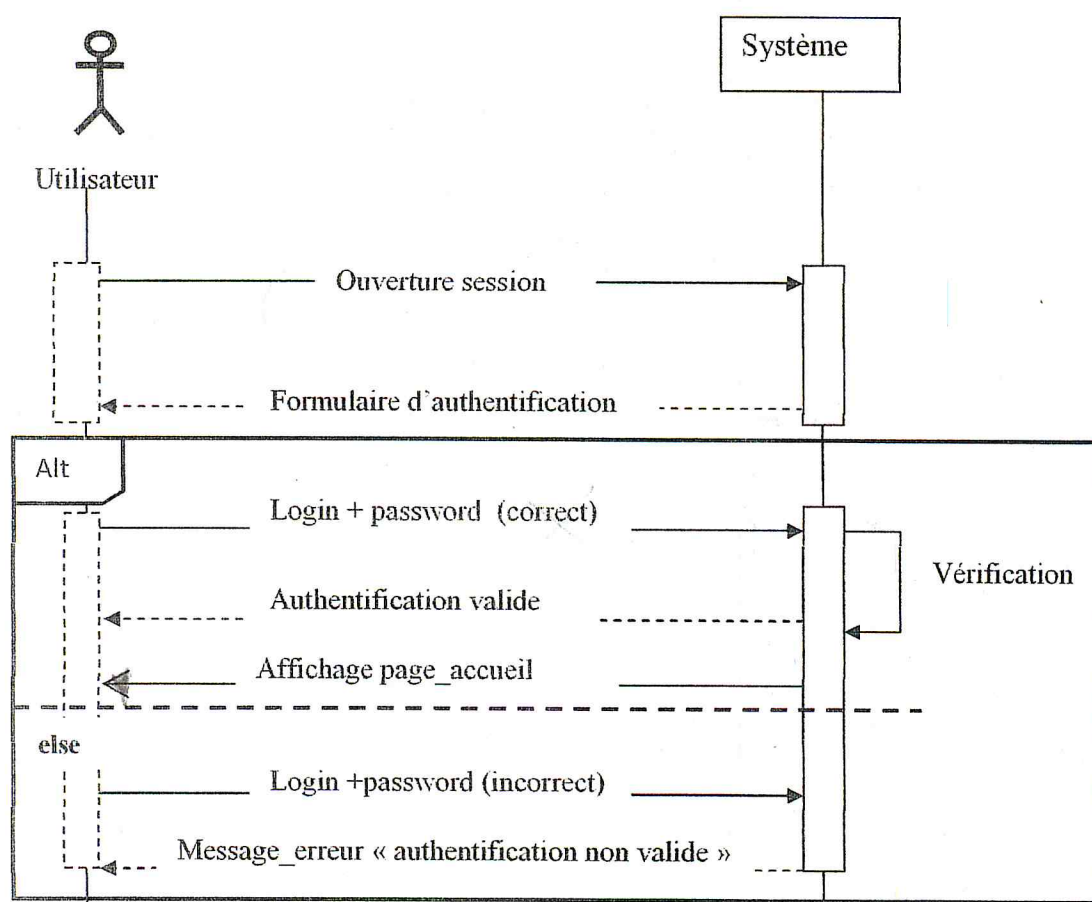
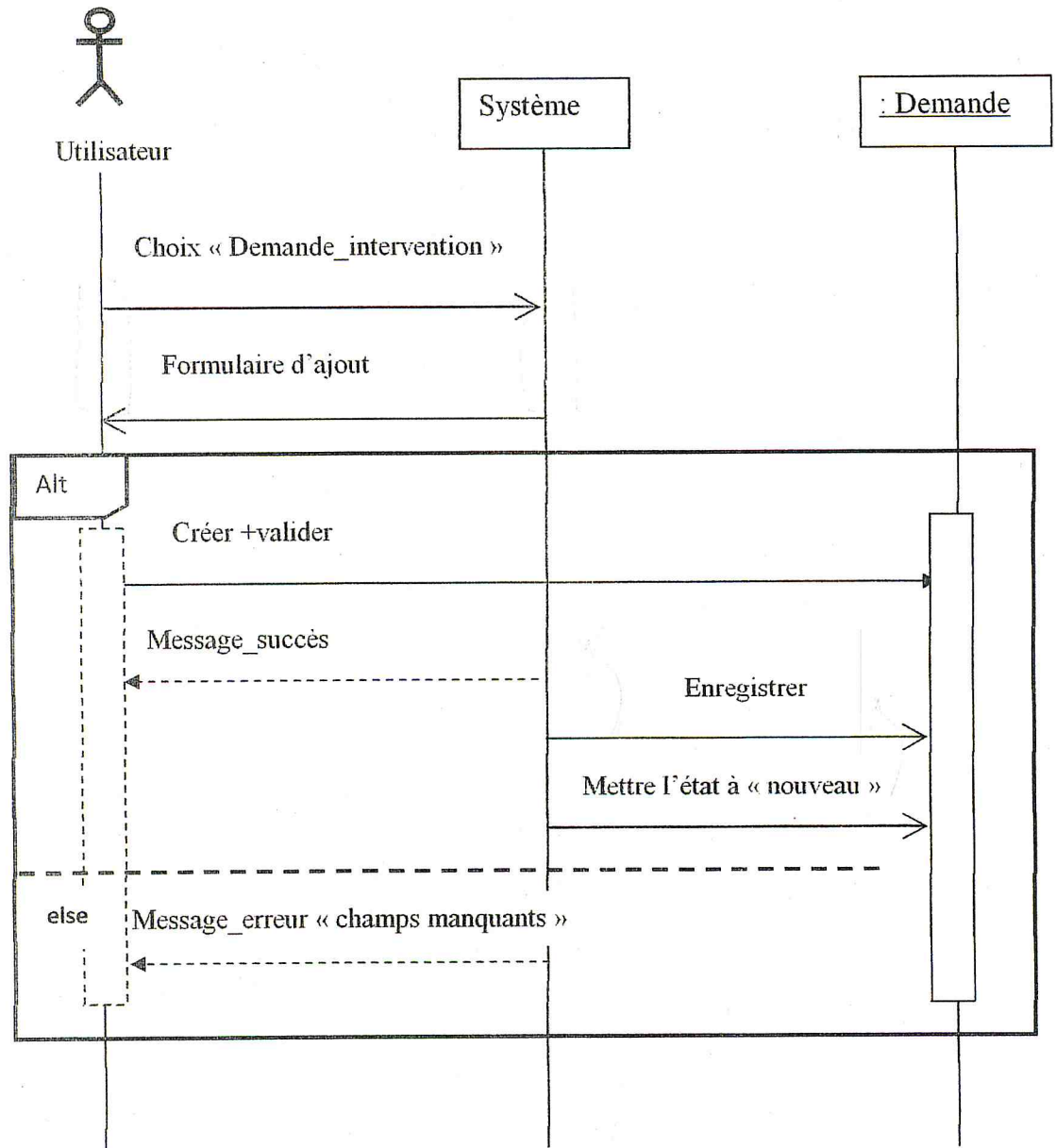


Figure 10 : Diagramme de séquence : « authentication »

3.2.1.2 Diagramme de séquence du cas d'utilisation « demander intervention »

*Figure 11 : Diagramme de séquence : « demander une intervention »*

3.2.1.3 Diagramme de séquence du cas d'utilisation « assigner un technicien » :

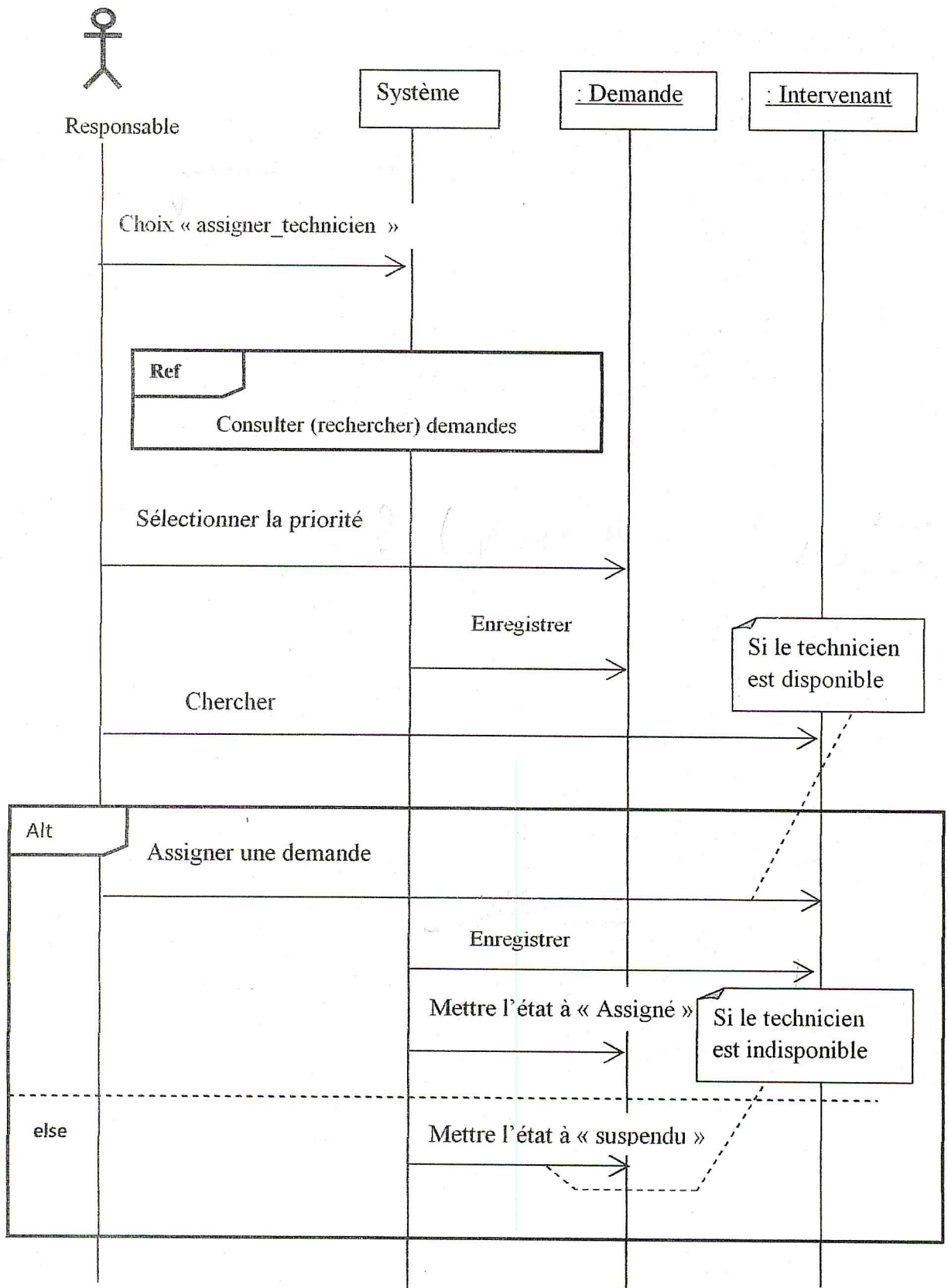
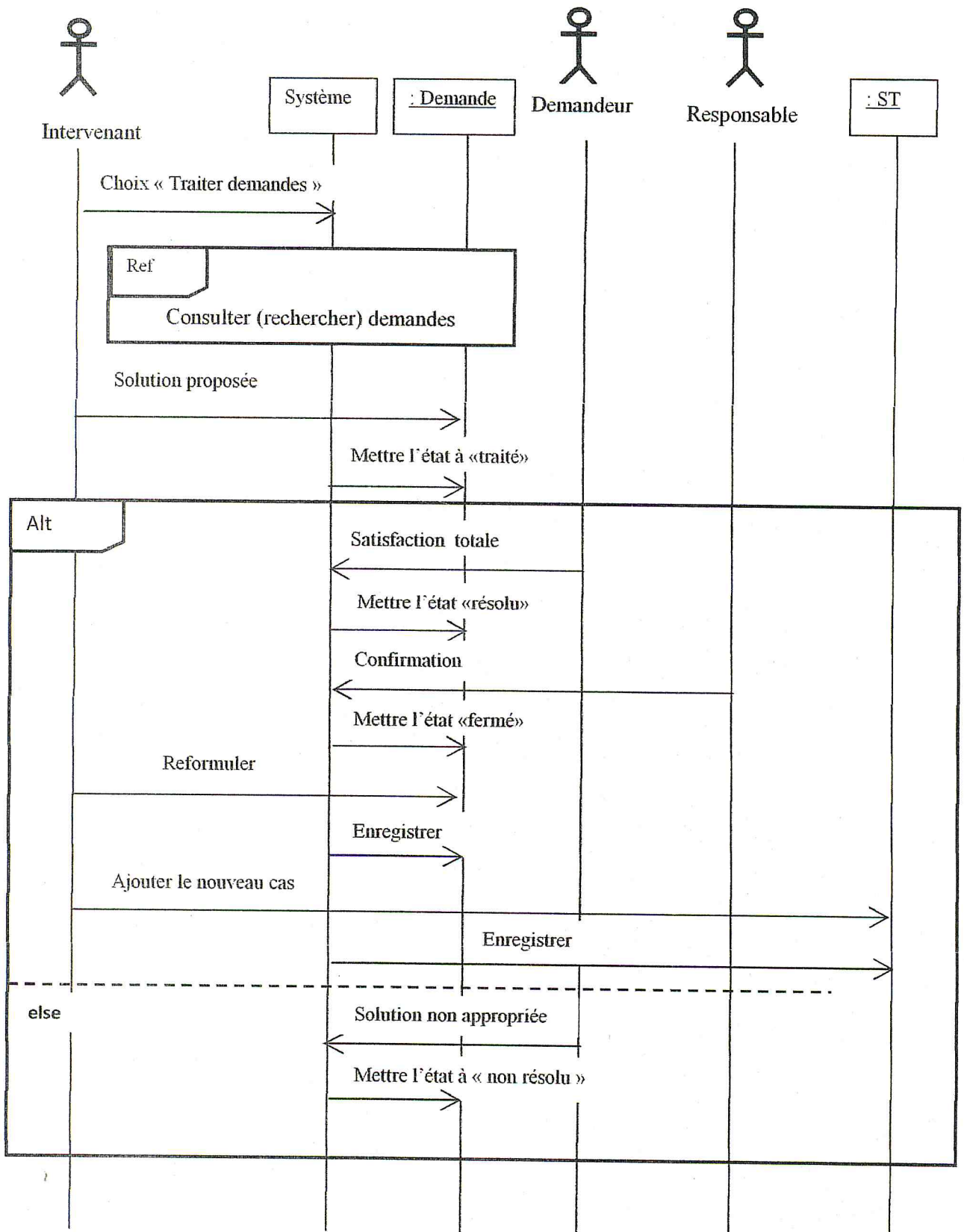


Figure 12 : Diagramme de séquence : « assigner un technicien »

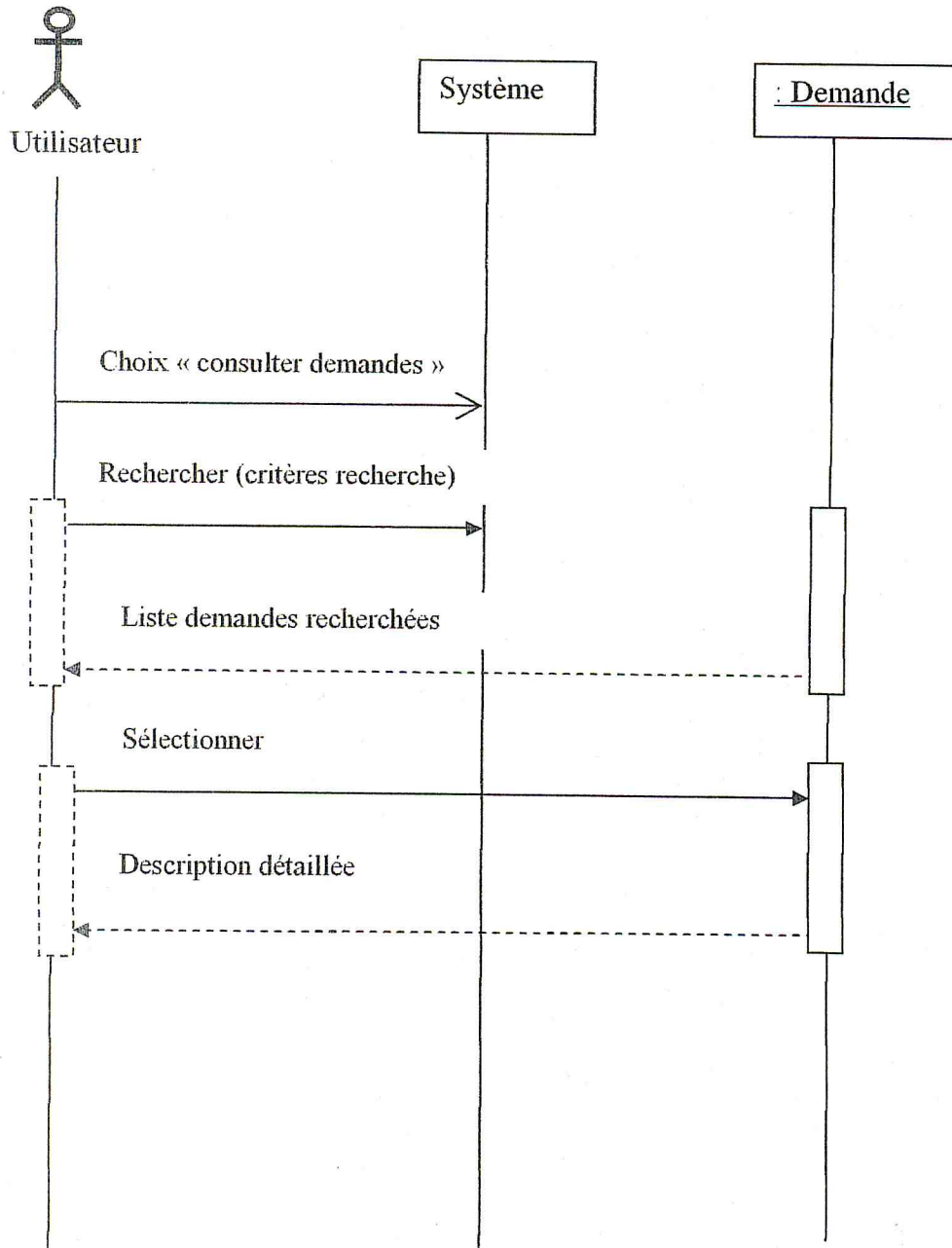


3.2.1.4 Diagramme de séquence du cas d'utilisation « traiter demande » :



*Figure 13 : Diagramme de séquence : « traiter la demande »*

### 3.2.1.5 Diagramme de séquence du cas d'utilisation «consulter/rechercher les demandes »



*Figure 14 : Diagramme de séquence : « consulter les demandes »*

3.2.1.6 Diagramme de séquence du cas d'utilisation « créer un compte » :

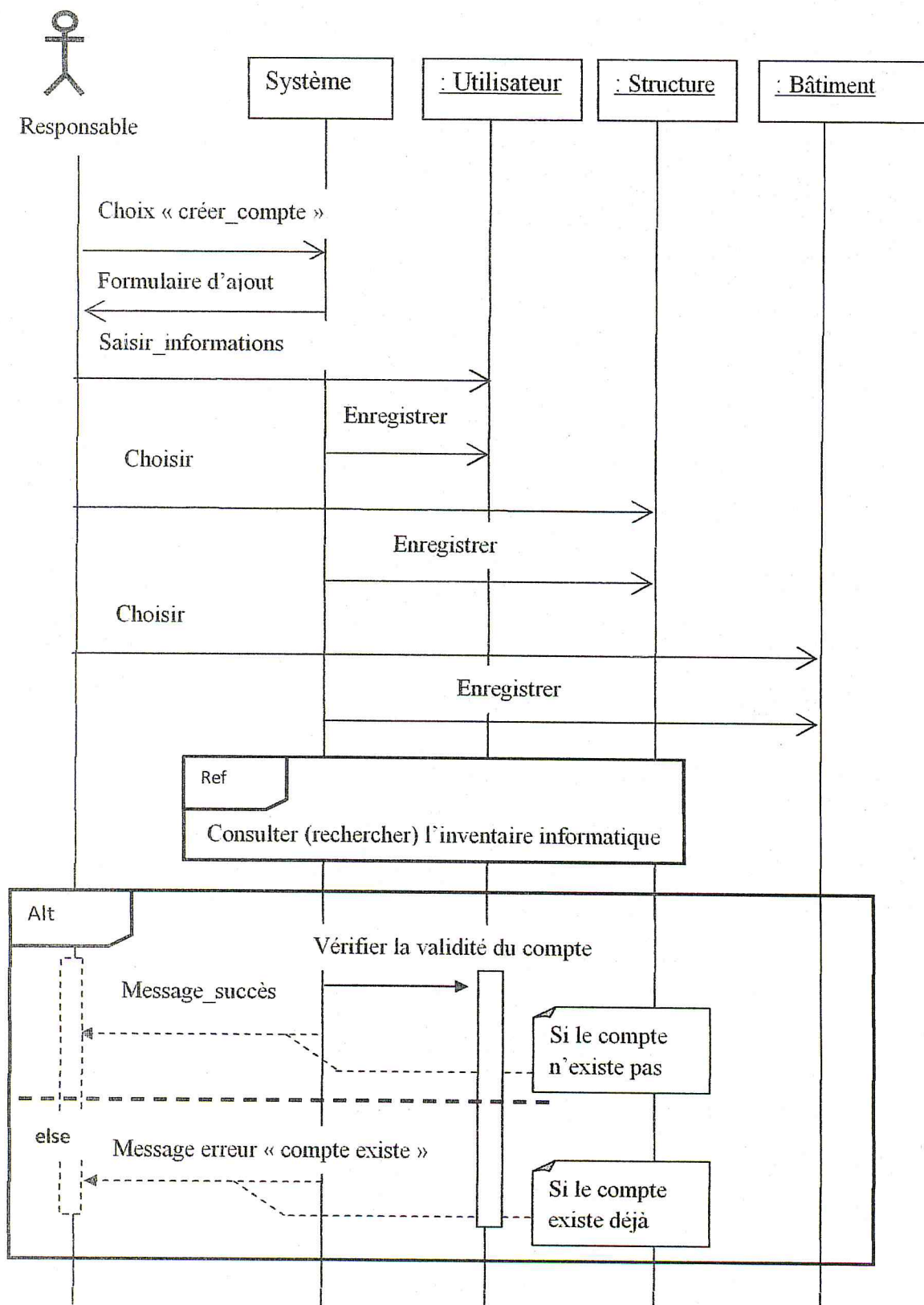
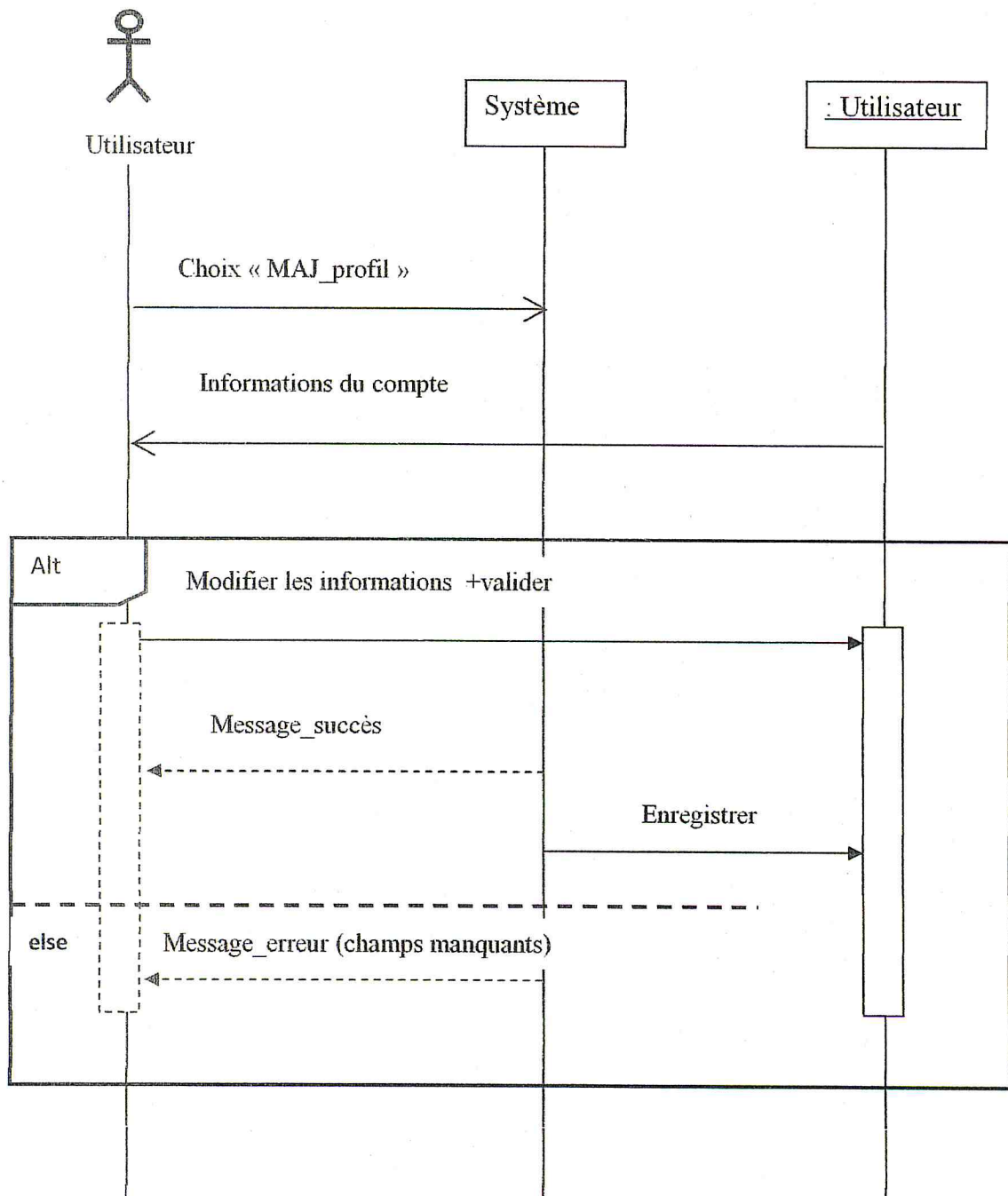
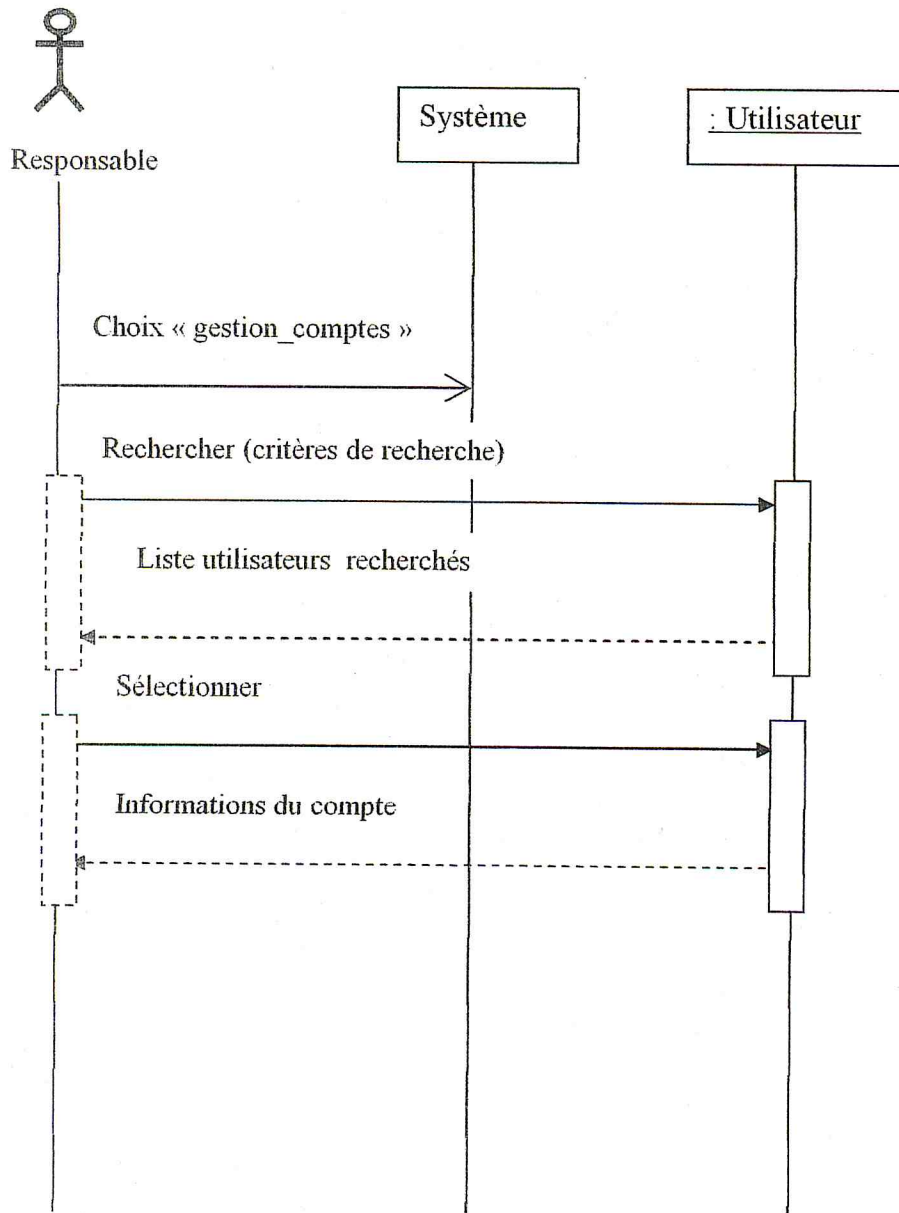


Figure 15 : Diagramme de séquence : « créer un compte »

**3.2.1.7 Diagramme de séquence du cas d'utilisation « modifier le profil »**

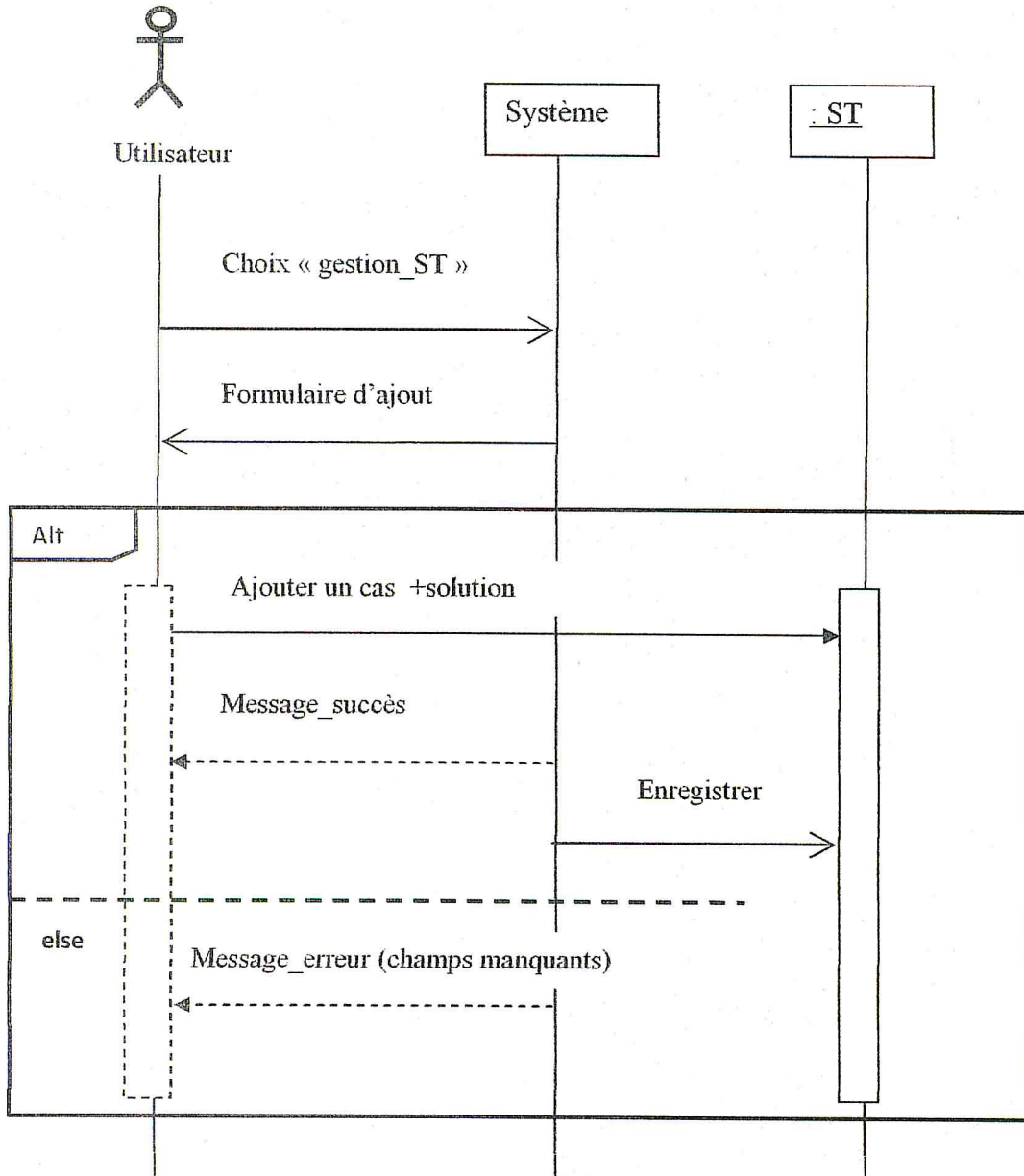
***Figure 16 : Diagramme de séquence : « modifier le profil »***



3.2.1.8 Diagramme de séquence du cas d'utilisation « consulter les comptes »

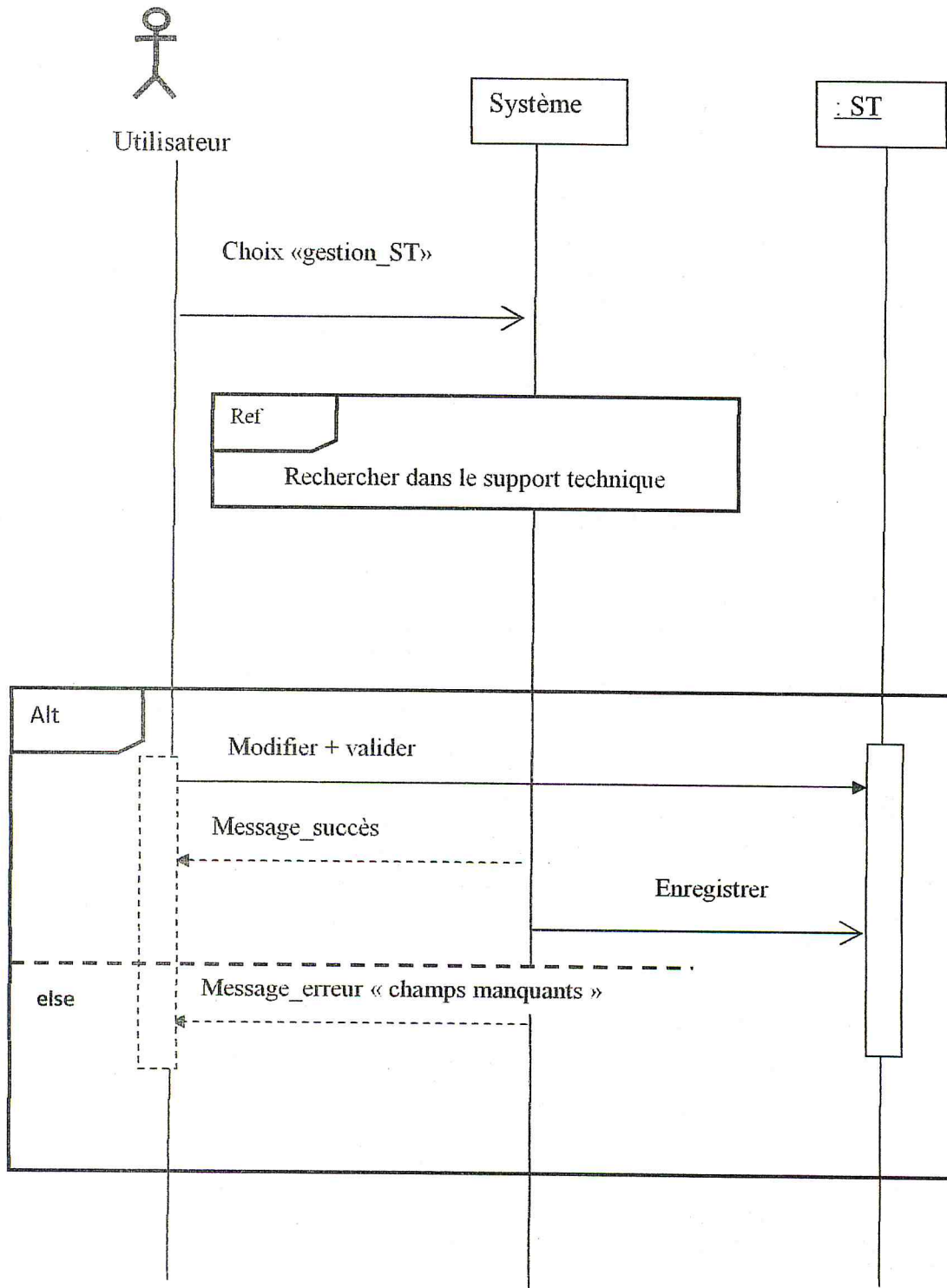
*Figure 17 : Diagramme de séquence : « consulter les comptes »*

**3.2.1.9 Diagramme de séquence du cas d'utilisation « ajouter un cas dans le support technique »**

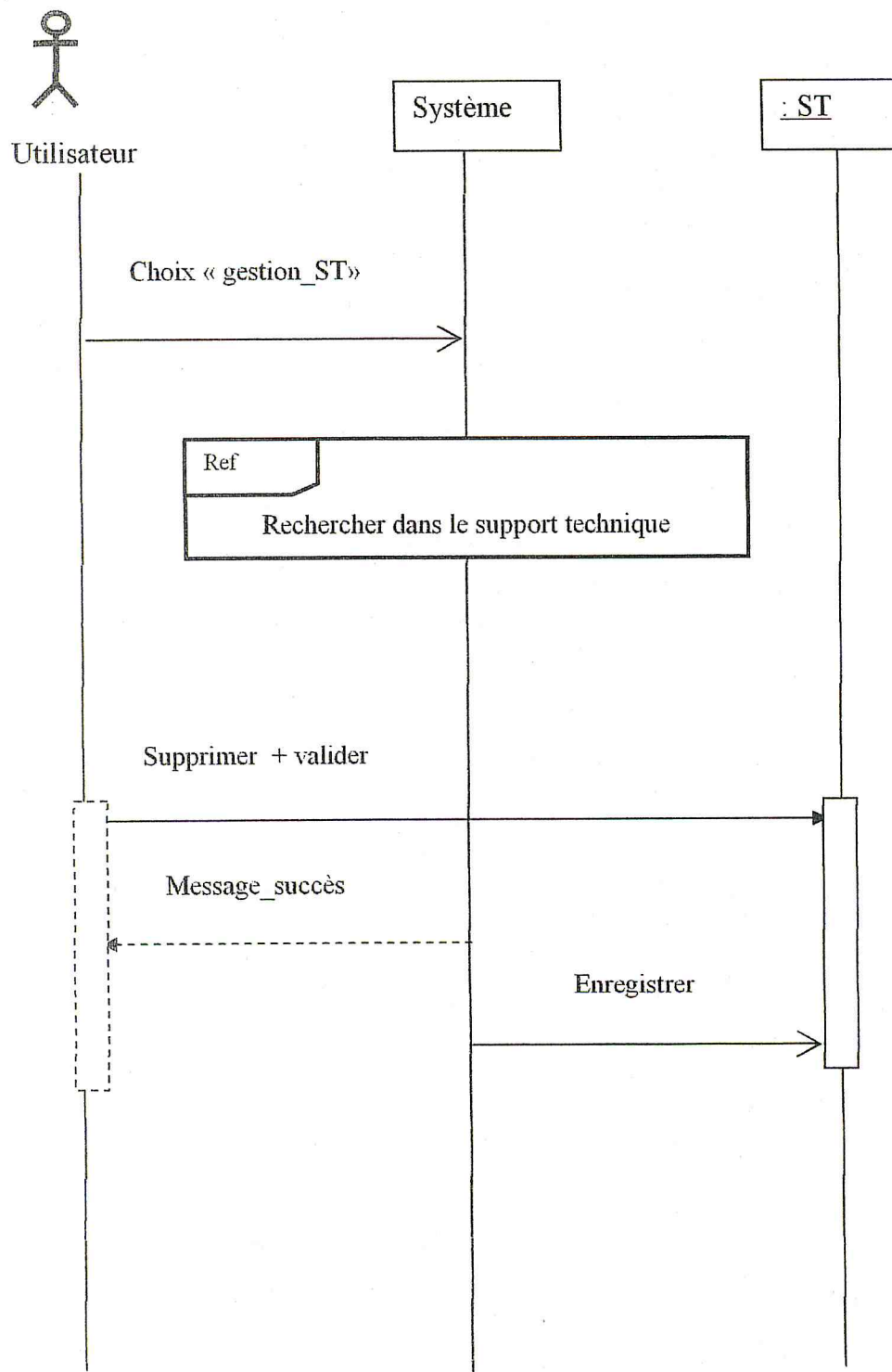


*Figure 18 : Diagramme de séquence : « ajouter un cas dans le support technique »*

**3.2.1.10 Diagramme de séquence du cas d'utilisation « modifier le support technique » :**



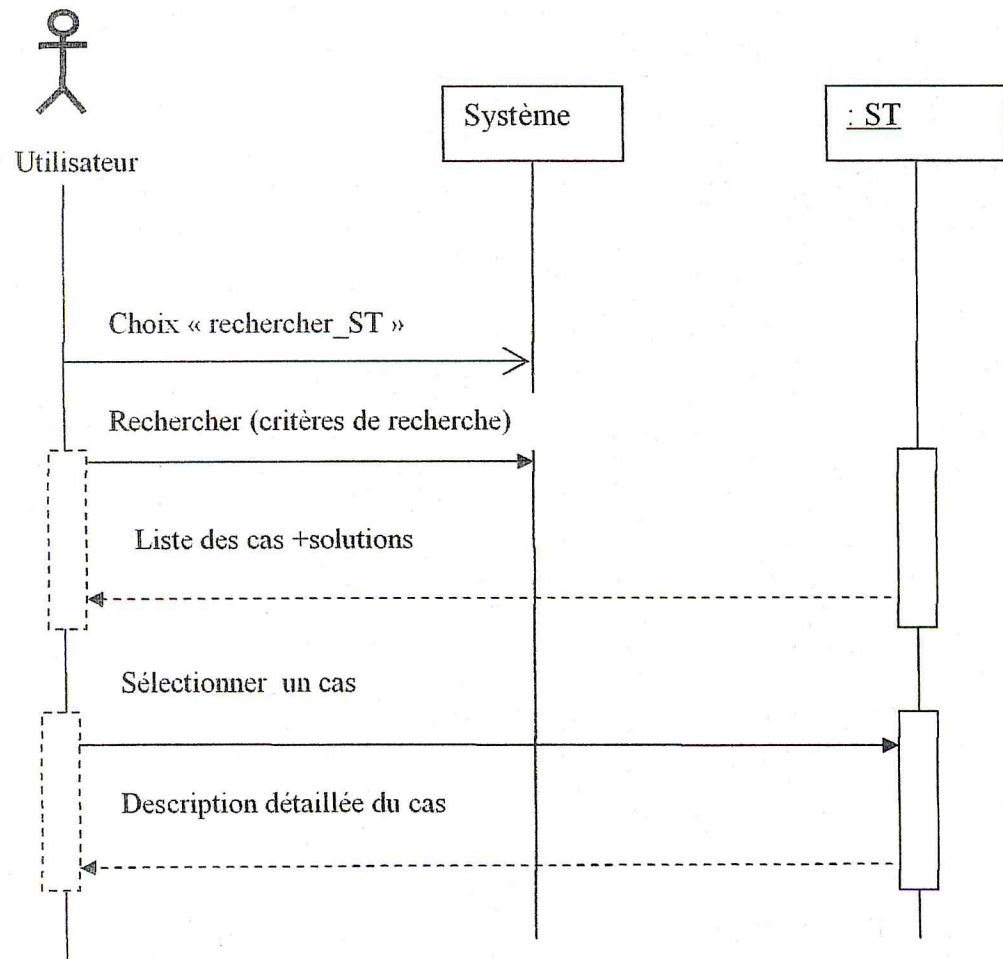
*Figure 19 : Diagramme de séquence : « modifier le support technique »*

**3.2.1.11 Diagramme de séquence du cas d'utilisation « supprimer un cas du support technique » :**

*Figure 20 : Diagramme de séquence : « supprimer un cas du support technique »*



**3.2.1.12 Diagramme de séquence du cas d'utilisation « rechercher dans le support technique » :**



*Figure 21 : Diagramme de séquence : « rechercher dans le support technique »*

3.2.1.13 Diagramme de séquence du cas d'utilisation « ajouter un équipement »

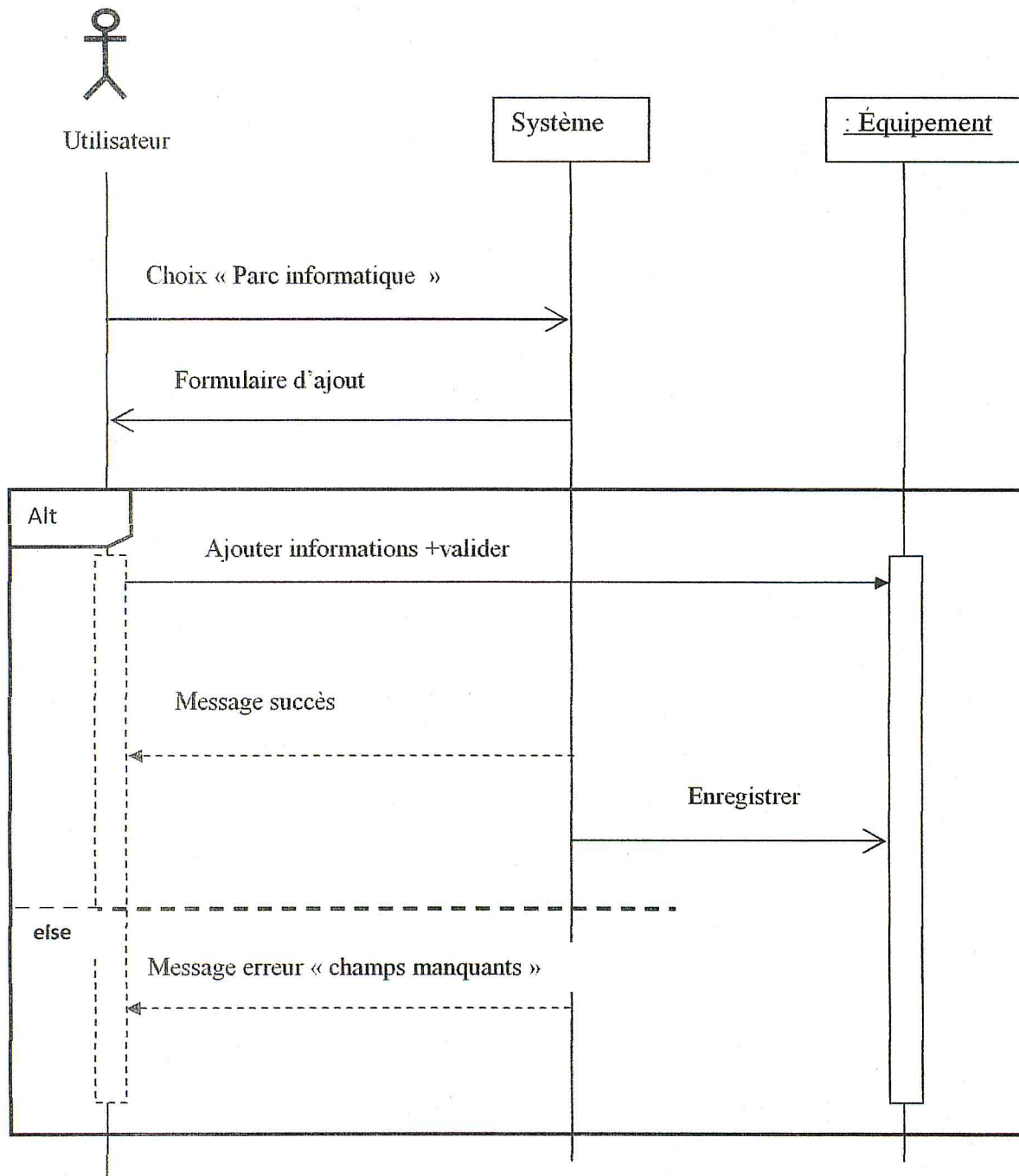
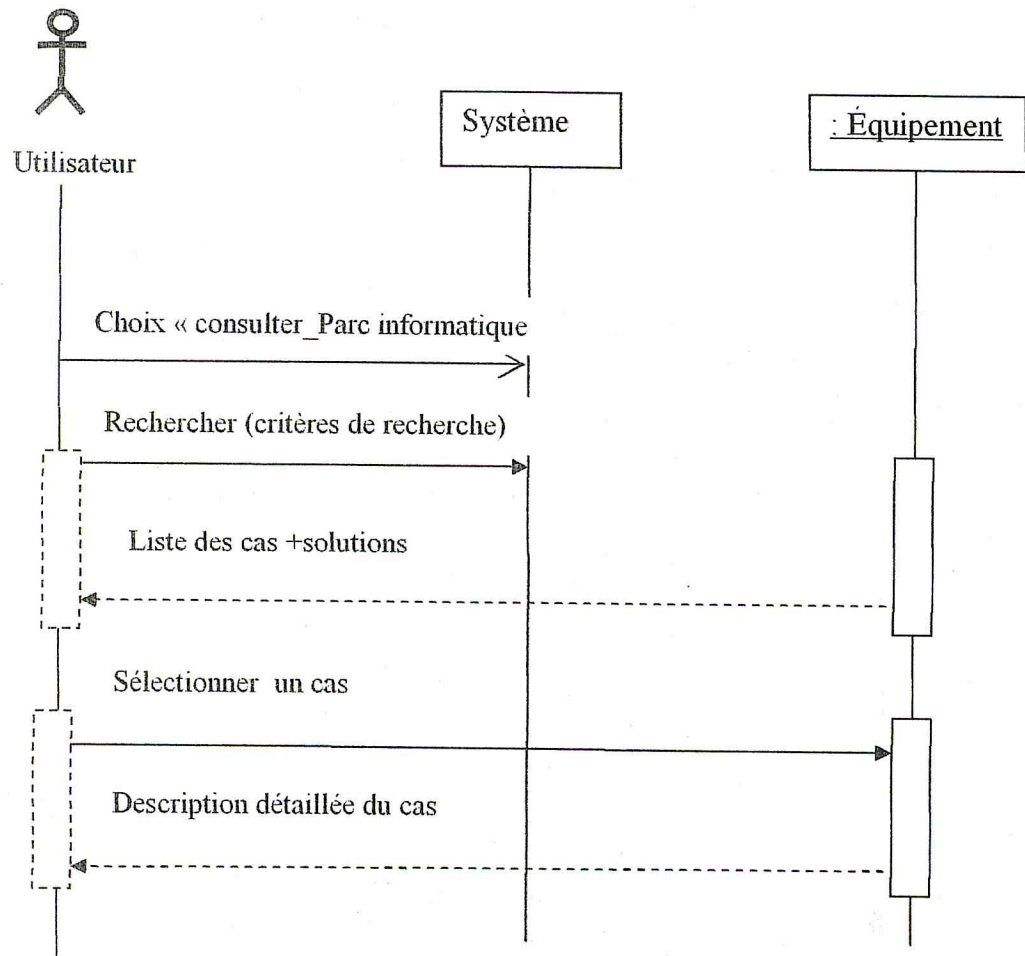


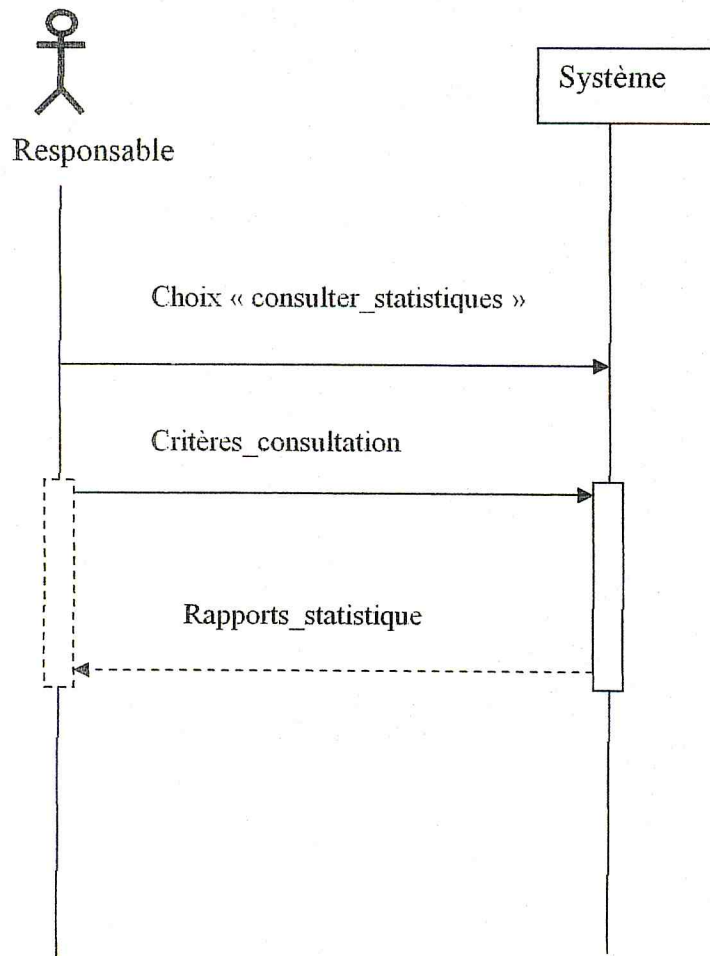
Figure 22 : Diagramme de séquence : « ajouter un équipement »

**3.2.1.14 Diagramme de séquence du cas d'utilisation « consulter (rechercher) l'inventaire informatique » :**



*Figure 23 : Diagramme de séquence : « rechercher dans le support technique »*

### 3.2.1.15 Diagramme de séquence du cas d'utilisation « consulter les rapports des statistiques »



*Figure 24 : Diagramme de séquence : « consulter les rapports des statistiques »*



### 3.3 La Conception

La conception permet de décrire de manière non ambiguë le fonctionnement futur du système, afin d'en faciliter la réalisation. La conception menée à la suite de la phase d'analyse répond donc à la question « *comment faut-il faire ce qu'il faut faire ?* ».

Ainsi, l'analyse sert à la découverte et à la compréhension et a pour but d'élaborer les spécifications tandis que la conception sert à structurer et à développer une solution

Dans cette phase, nous utilisons le diagramme de classes pour montrer l'architecture du système, et le diagramme d'états-transition pour exprimer les différents états d'un objet d'une classe. *[Proc]*

#### 3.3.1 Diagramme de classes

Le diagramme de classe est considéré comme le plus important de la modélisation orienté objet. Alors que le diagramme de cas d'utilisation montre un système de point de vue des acteurs, le diagramme de classes en montre la structure interne. Il contient principalement des classes. Une classe contient des attributs et des opérations.

Le diagramme des classes n'indique pas comment utiliser les opérations : c'est une description purement statique d'un système.

#### Qu'est qu'une classe ?

Une classe est une description d'un ensemble d'objets ayant une sémantique, des attributs, des méthodes et des relations en commun. *[UML2]*

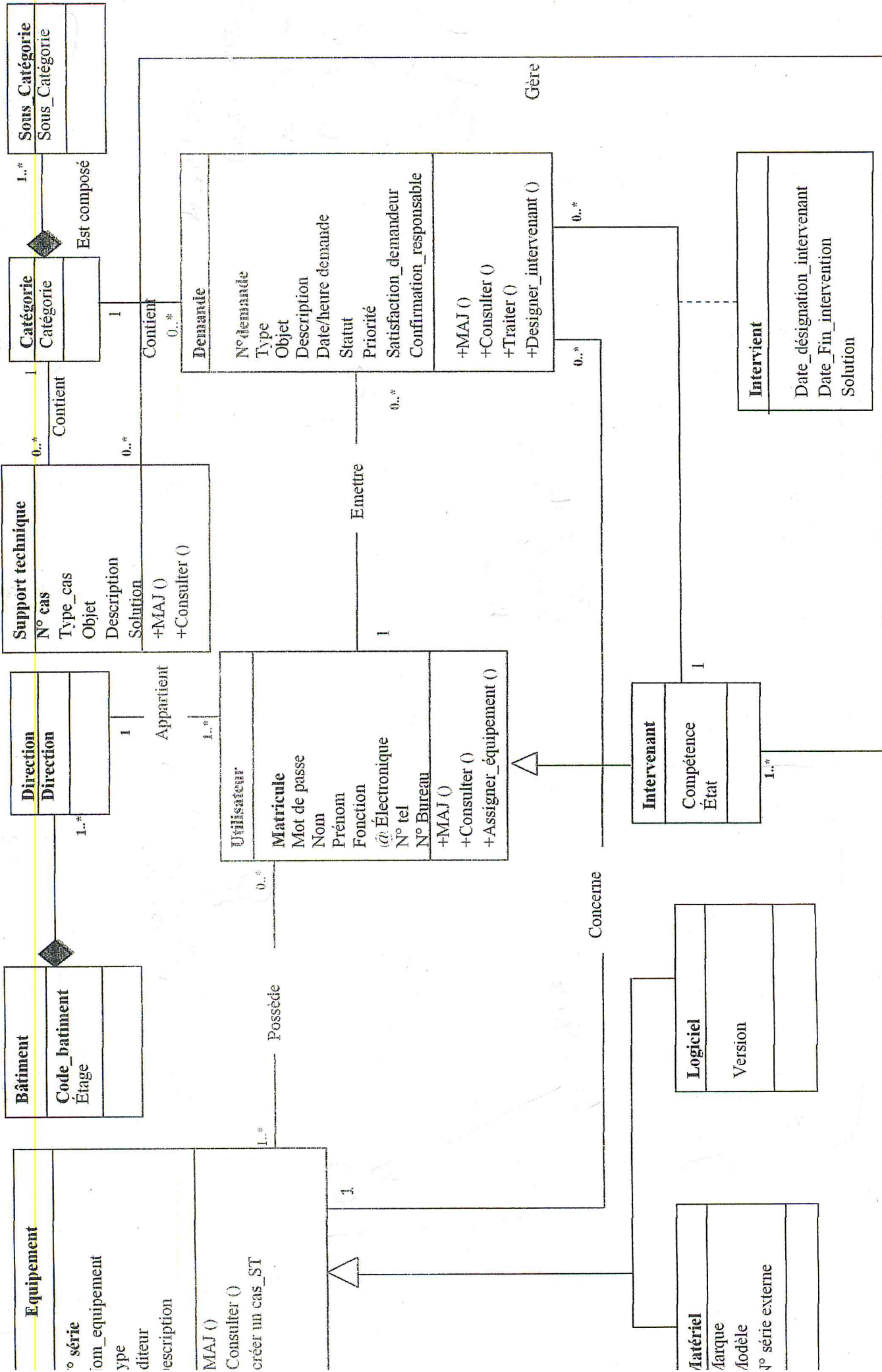


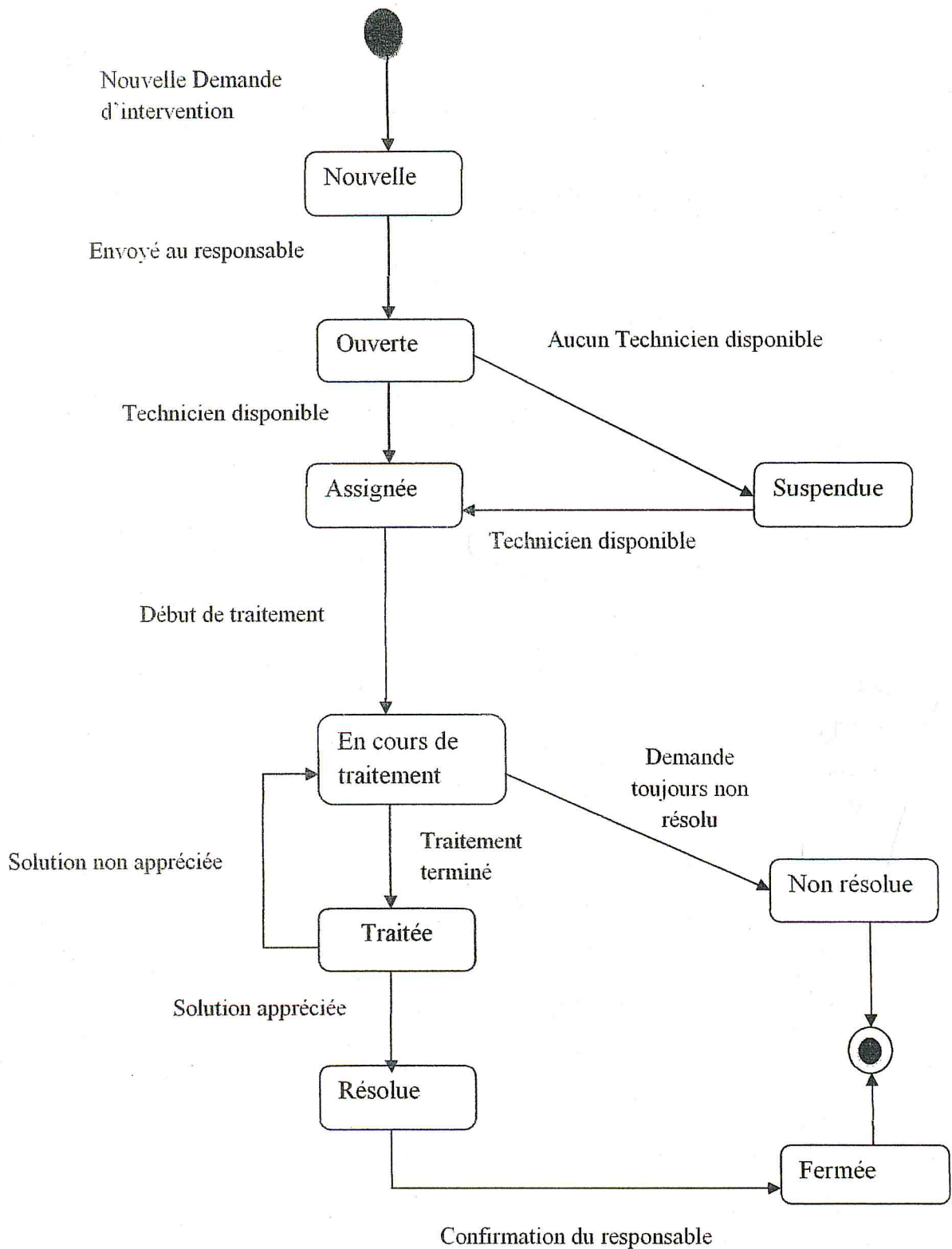
Figure 25 : Diagramme de classes

### 3.3.2 Diagramme d'états-transitions :

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets (de type signaux, invocations et méthodes). Ils spécifient habituellement le comportement d'une instance de classe, mais parfois aussi le comportement interne d'autres éléments tels que les cas d'utilisation, les méthodes. Ils sont bien adaptés à la description d'objets ayant un comportement d'automate. [UML2]

Le traitement des demandes d'utilisateurs passe par plusieurs états :

- Nouvelle : C'est l'état initial de la demande, lors de sa création par le demandeur
- Ouverte : la demande est envoyée et ouverte par le responsable.
- Assignée : montre que le technicien est disponible.
- Suspendu : en cas où le technicien n'est pas disponible, le système met l'état de la demande à « suspendu ».
- En cours de traitement : Indique que le technicien traite la demande envoyée par le responsable.
- Traité : le traitement de la demande est accompli et envoyée au demandeur concerné
- Résolu : si la solution proposée par le technicien satisfait le demandeur, l'état de la demande devient « résolu ». Par contre, si la solution ne lui convient pas, le technicien retraite la demande à nouveau, ou bien le responsable attribue un autre technicien.
- Fermé : le technicien envoi le diagnostic de son intervention au responsable, qui met l'état de la demande à « fermé »



*Figure 26 : Diagramme d'états-transition*



*L'analyse faite à travers ces différents diagrammes a permis d'effectuer une spécification plus détaillées des besoins et nous a servi de base pour les mécanismes internes du système.*

*Ceci constitue une entrée essentielle pour la réalisation de l'application, que nous allons présenter dans le prochain chapitre.*

## ***IV. L'implémentation***

---

### **L'objectif du chapitre :**

C'est la phase de réalisation et de mise en œuvre, avec le déploiement d'outils de développement logiciel qui correspondent à l'architecture du système.

### **Dans ce chapitre :**

- SQL\*Plus
- PL/SQL.
- Oracle Application Server.
- Forms Builder.
- Oracle Reports.

### **Éléments mis en jeu :**

- ✓ SQL Plus, PL/SQL.
- ✓ Oracle Developer.
- ✓ Oracle Forms, Reports.
- ✓ Interface.

*Pour l'implémentation et la construction de notre application help desk selon une architecture 3tiers, nous avons choisi parmi les outils dont dispose le département de développement informatique, ceux qui dérivent des produits oracle (dans sa version 10g) et cela, par rapport à sa performance et son efficacité qui sont reconnus dans le monde du développement informatique.*

### **1. Les outils d'implémentation :**

Oracle propose de nombreux outils de développement permettant d'automatiser la création d'applications s'interfaçant avec la base de données. Ces outils de développement sont nombreux : SQL\*Plus, Oracle Designer, Oracle Développer...etc.

Pour le développement de notre application nous avons choisi :

- SQL\*Plus pour la gestion de la base de données ;
- Oracle Application Server qui est un serveur d'application très complet. Il dispose de nombreux composants qui lui permettent de fournir de multiples services aux utilisateurs ;
- Forms Builder est un outil qui permet de créer et de déployer des applications multi plates-formes fonctionnant dans un navigateur à l'aide d'une applet Java spécifique.

#### **1.1. SQL\*Plus :**

SQL\*Plus est prévu pour définir et manipuler les données dans une base de données relationnelle Oracle. Le langage SQL (Structured Query Language, langage de requête structuré) est le standard de l'industrie adopté par tous les éditeurs de base de données. Le langage SQL\*Plus d'Oracle est un surensemble de ce standard. Il respecte les spécifications d'un langage compatible SQL, mais intègre certaines extensions propres à Oracle, d'où son nom SQL\*Plus (anciennement appelé UFI, User-Friendly Interface, interface conviviale). Comparé au développement avec un langage de programmations tel que FORTRAN, on peut effectivement dire qu'il s'agit là d'une interface conviviale. [Ora1]

### 1.2. PL/SQL :

Depuis la version 6, Oracle a implémenté un langage de traitement procédural appelé PL/SQL qui facilite grandement le traitement des tâches. Les structures de ce langage sont semblables à celles de la plupart des autres langages de programmation.

Il existe deux versions de PL/SQL : l'une fait partie du moteur de base de données, l'autre est un moteur séparé imbriqué dans un certain nombre d'outils Oracle. Elles sont appelées respectivement PL/SQL base de données et PL/SQL outil. Toutes deux utilisent les mêmes structures, syntaxe et logique de programmation. PL/SQL outil supporte une syntaxe additionnelle lui permettant de gérer les exigences des outils.

Par exemple, le placement d'un bouton sur un formulaire pour naviguer vers le bas de l'écran sera codé au moyen de PL/SQL dans Oracle Forms. *[Ora1]*

### 1.3. Oracle Application Server

Oracle commercialise un serveur d'application appelé Oracle Application Server. Il utilise un ORB (Object Request Broker) pour gérer l'interaction entre les clients et les serveurs.

A l'instar de la technologie client-serveur, plusieurs composantes sont nécessaires à l'utilisation d'OAS. Certaines de ces composantes sont fournies par Oracle ; celles qui ne sont pas fournies sont marquées d'un astérisque (\*) :

- Un navigateur Internet (\*).
- Un listener http, qui détecte les demandes entrantes (Oracle fournit le listener http Spyglass).
- Des cartouches (cartridges), pour exécuter les applications et retourner le contenu HTML. Les cartouches sont des produits logiciels – proposés par Oracle ou des fabricants tiers- qui communiquent avec la base de données en utilisant les appels OCI (Oracle Call Interface) ou PL/SQL.
- Un dispatcher, pour les requêtes qui nécessitent des cartouches.
- Le médiateur Web Request Broker (WRB).
- Une base de données Oracle (déjà construite). *[Ora2]*



### 1.4. Forms Builder :

Après avoir créé une puissance base de donnée (SQL\*Plus), il apparaissait tout naturel qu'Oracle souhaite fournir un générateur d'écrans, puisque c'est le principal moyen d'interaction des utilisateurs avec la base de données. Oracle Forms Builder fait partie de la suite d'outils Oracle Developer et représente l'interface utilisateur de choix pour les applications critiques de l'entreprise, il s'exécute dans un environnement graphique et adopte le même aspect que Microsoft Windows.

Oracle forms permet aux développeurs d'applications de créer des écrans de saisie et de requêtes de données que les utilisateurs finaux peuvent employer pour manipuler les informations dans la base d'Oracle. L'interface présentée aux utilisateurs est guidée par des événements. *[Ora1]*

### 1.5. Oracle Reports :

Bien que SQL\*Plus soit un outil de reporting efficace, il n'a pas été conçu dans cet objectif. Oracle a développé plusieurs outils spécifiques tels que SQL\*QMX, RPT et Easy\*SQL. Oracle Reports est l'utilitaire de génération d'états phare de la suite Developer.

Oracle Reports succède à SQL\*Report Writer qui accompagnait la version 6 d'Oracle et proposait un environnement permettant aux développeurs de concevoir des rapports et aux utilisateurs de les exécuter. Il s'est transformé en Oracle Reports à mesure que les utilisateurs insistaient pour bénéficier d'une interface de type Windows faisant appel à la souris. Avec Oracle Reports, il est possible de créer des représentations graphiques d'états à partir des données contenues dans la base Oracle. *[Ora1]*

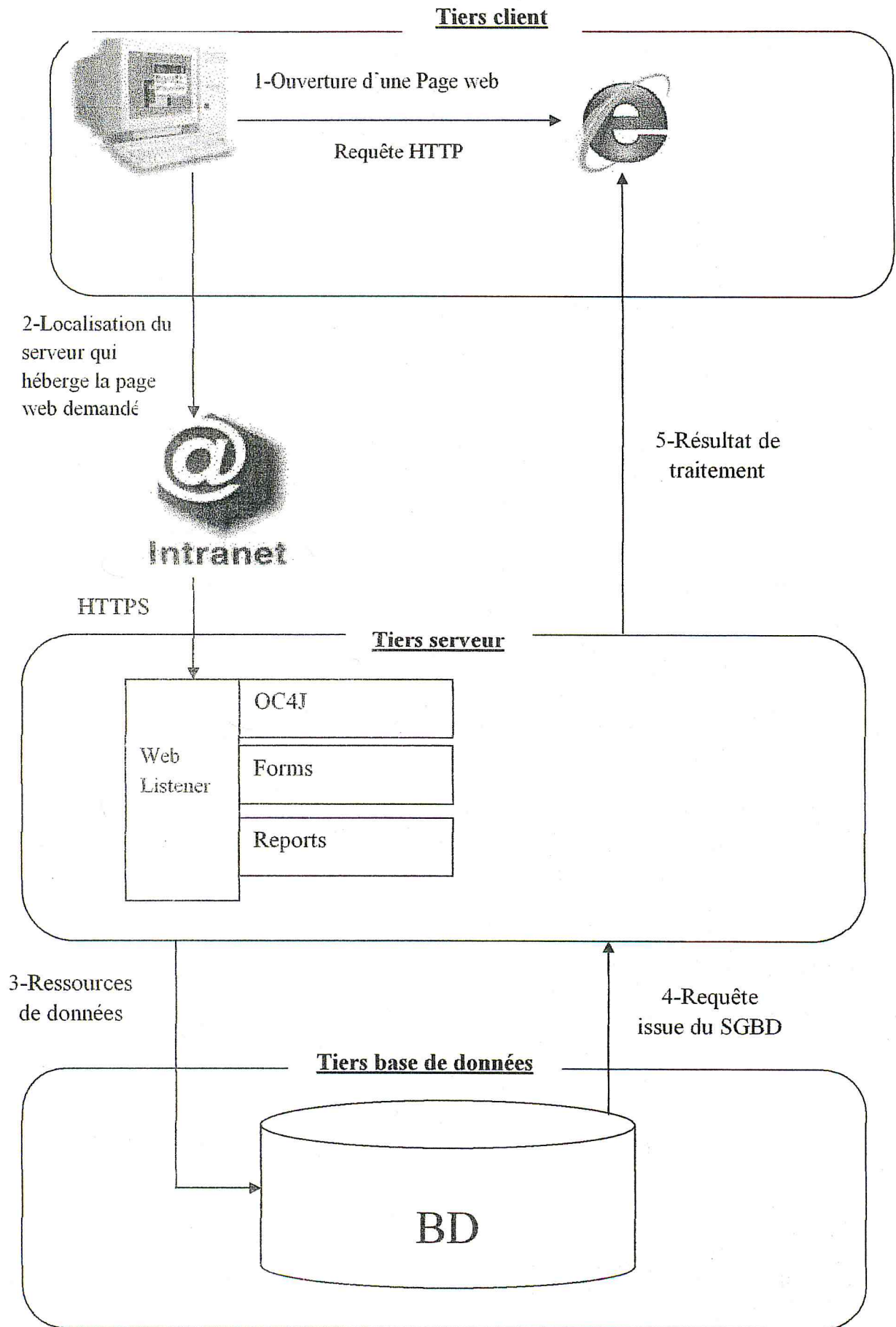


Figure27 : Architecture 3 tiers



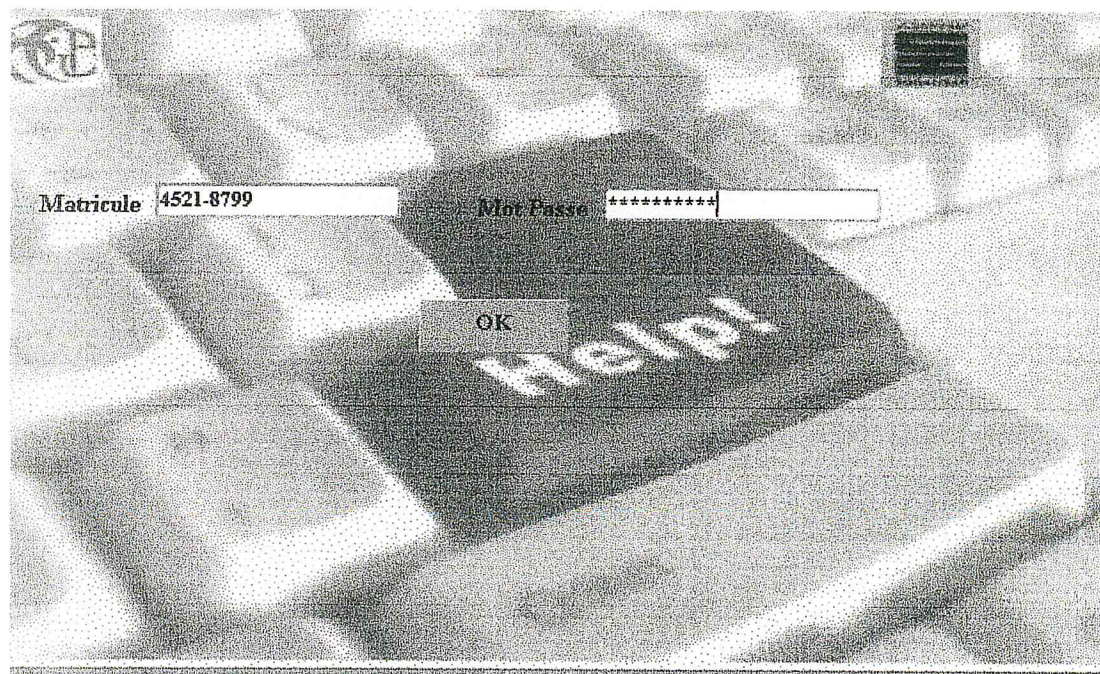
## 2. Réalisation et mise en œuvre de l'application :

Notre application peut être accédée par l'utilisateur à travers le portail d'entrée représenté par la page d'accueil. La mise en œuvre de l'application consiste donc à mettre en place les meilleures conditions pour l'environnement destiné à l'utilisateur final de l'application.

Les fenêtres sont représentées dans les pages suivantes :

### 2.1. Fenêtre d'authentification

Cette interface permet aux différents utilisateurs d'accéder à la page d'accueil de Help Desk, selon leurs privilèges



*Figure28 : fenêtre d'authentification*



### 2.2. L'ajout d'une nouvelle demande d'intervention :

Grâce à cette interface, le demandeur peut solliciter l'intervention d'un technicien, en décrivant tous les renseignements de sa demande :

The screenshot shows a software window titled 'Demande d'intervention' with a menu bar containing 'Demande d'intervention', 'Consultation des demandes', 'Traitement Demandes', 'Assignment technicien', and 'Historique d'intervention'. The main area contains the following fields:

- Type:** ASSISTANCE
- Catégorie:** LOGICIEL
- Sous catégorie:** VIRUS
- Objet:** blocage d'ordinateur
- Description:** bonjour  
mon ordinateur ne marche pas  
il est trop lent au démarrage  
merci de m'aider

An 'OK' button is located at the bottom center of the window.

*Figure29 : fenêtre de création d'une nouvelle demande*

### 2.3. L'assignation d'un technicien :

Le responsable assigne le technicien approprié au type de la demande posée

The screenshot shows a software window titled 'Assignment technicien' with a menu bar containing 'Demande d'intervention', 'Consultation des demandes', 'Traitement Demandes', 'Assignment technicien', and 'Historique d'intervention'. The main area is divided into two sections:

**Demande**

- N°:** 1
- Type:** ASSISTANCE
- Catégorie:** LOGICIEL
- Sous catégorie:** VIRUS
- Date demande:** 10/06/2010
- Objet:** blocage d'ordinateur
- Description:** bonjour  
mon ordinateur ne marche pas  
il est trop lent au démarrage  
merci de m'aider
- Statut:** nouveau
- Priorité:** Urgent

**Assignment Technicien**

- Matricule:** 456-987
- Technicien:** Hamizi
- Compétence:** Ingénieur
- Etat:** Disponible

Buttons for 'Rechercher' and 'Assigner' are visible.

*Figure30 : fenêtre d'assignation d'un technicien*



## 2.4. Traitement des demandes :

Le technicien propose une solution, et l'envoie au demandeur concerné, si la demande est résolue (satisfaite par le demandeur), il l'ajoute au support technique.

Demande			
N°	1	Type	ASSISTANCE
Catégorie	LOGICIEL	Sous catégorie	VIRUS
Date demande	10/06/2010		
Objet	blocage d'ordinateur		
Description	bonjour mon ordinateur ne marche pas il est trop lent au démarrage merci de m'aider		
Statut	nouveau	Priorité	Urgent
Intervention			
Date désignation	10/06/2010	Date fin d'intervention	10/06/2010
Solution proposée	analysez votre ordinateur avec un antivirus efficace et supprimez tous les virus		
Envoyer			
satisfaite par le demandeur <input type="checkbox"/>			

Figure 31 : fenêtre de traitement des demandes

## 2.5. Création d'un compte

Le responsable crée les différents comptes d'utilisateur, et assigne les équipements possédés.

Créer compte			
Type d'utilisateur	Demandeur		
Matricule	456-789		
Nom	Hamizi	Prénom	Soumia
Mot de passe	*****	Fonction	Gestionnaire
@elect	abc@yahoo.fr	Telephone	012345678
N° Bureau	85	Direction	Achat & Stock
Batiment	A	étage	2
Assigner équipement			
Créer			

Figure 32 : fenêtre de création d'un compte



## 2.6. MAJ le profil

Cette interface permet à l'utilisateur de modifier ses paramètres personnels

Créer compte Profil Consulter

Matricule 4521-8799

Changement de mot de passe

Ancien mot.passe \*\*\*\*\* Nouveau mot\_passe \*\*\*\*\*

Confirmez mot\_passe \*\*\*\*\*

OK

Nom Nehab Prenom wissam

Fonction Gestionnaire @Electronique aaaa@yahoo.fr

Telephone 1236547 Bureau 11

Batiment C Etage 3

Direction Finance

Enregistrer

*Figure33 : fenêtre de MAJ de profil*

## 2.7. Ajouter un équipement

Ajouter MAJ

N° serie 313-547 Nom.Equipement imprimante

Type Matériel Description imprimante jet d'ancre

Fabriquant

Logiciel

Version

N° serie externe Sona-754-00

Marque HP Modèle Deskjet 3845

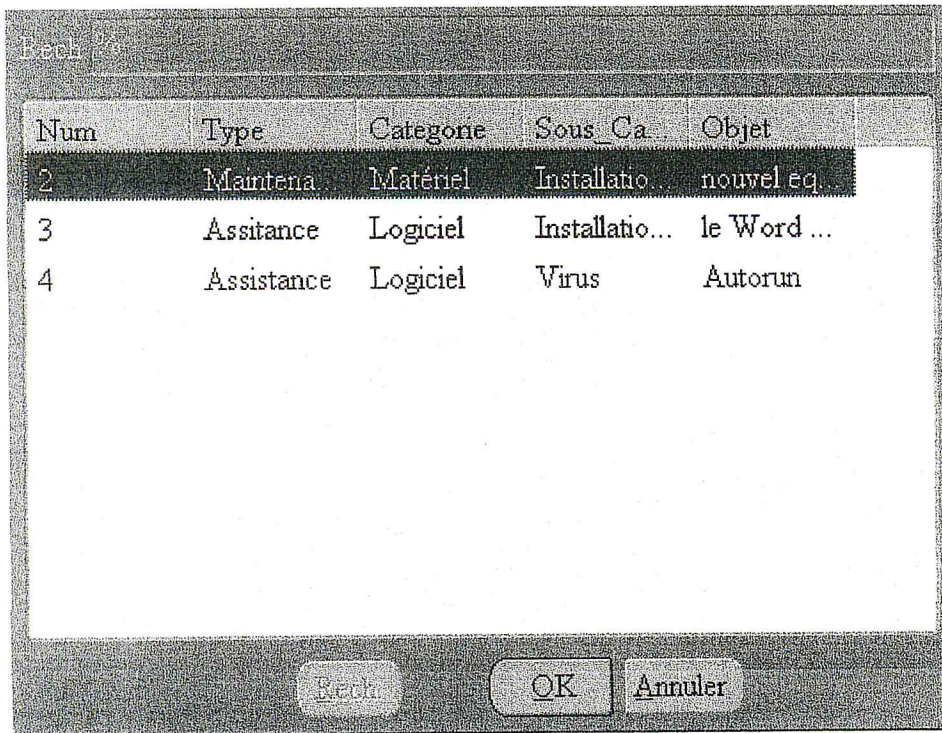
Ajouter

*Figure34 : fenêtre d'ajout d'un équipement*



**2.8. Support technique :**

Les utilisateurs peuvent consulter le support technique selon des critères de recherche bien définis



Num	Type	Catégorie	Sous_Ca	Objet
2	Maintena	Matériel	Installatio	nouvel eq.
3	Assistance	Logiciel	Installatio...	le Word ...
4	Assistance	Logiciel	Virus	Autorun

*Figure35 : fenêtre de la recherche dans le support technique*

*Dans cette partie, nous avons présenté l'environnement de développement de notre application Help Desk, qui cible à apporter une solution et répondre aux besoins des utilisateurs avec une manipulation facile et des interfaces graphiques ergonomiques, cette phase correspond à l'étape codage et test de RUP.*



## *V. Conclusion générale et perspectives*

---

Aujourd'hui, le help desk est devenu une application réactive, constamment dans un état de développement, et à travers ce travail, nous espérons améliorer l'efficacité des services fournis pour la gestion des incidents informatique, car la solution proposée, permettra en premier lieu d'assurer la maîtrise du processus d'intervention, de sauvegarder les inventaires informatique et utilisateurs, de fournir un point d'accès unique pour la prise en charge des demandes de tous les utilisateurs, et d'accroître la qualité du service fourni par le département informatique, avec la possibilité d'évaluer et d'analyser l'activité maintenance et d'anticiper les solutions .

En utilisant ce Help desk, Chaque poste client disposera d'une interface utilisateur lui permettant de rédiger et de transmettre ses demandes d'interventions en fonction de la catégorie d'incident, et de suivre l'évolution de sa prise en charge.

Bruton explique que « *le rôle du Support va bien plus loin que simplement résoudre les problèmes techniques quand ils se produisent, Le but du helpdesk est de rendre les utilisateurs plus productifs* ». Précisons toutefois que si on veut qu'un help desk soit mis en place avec une fonction décisionnelle, cela demanderait du temps et de la ressource humaine. En effet, les informations doivent être enregistrées régulièrement par les analystes help desk dans la base de données. Par la suite, cela nécessiterait des efforts réguliers, tel que l'observation et l'analyse par le responsable du help desk.

Le service help desk nous a donné l'occasion de voir quelles sont les inquiétudes et les véritables problèmes auxquels les utilisateurs font face.

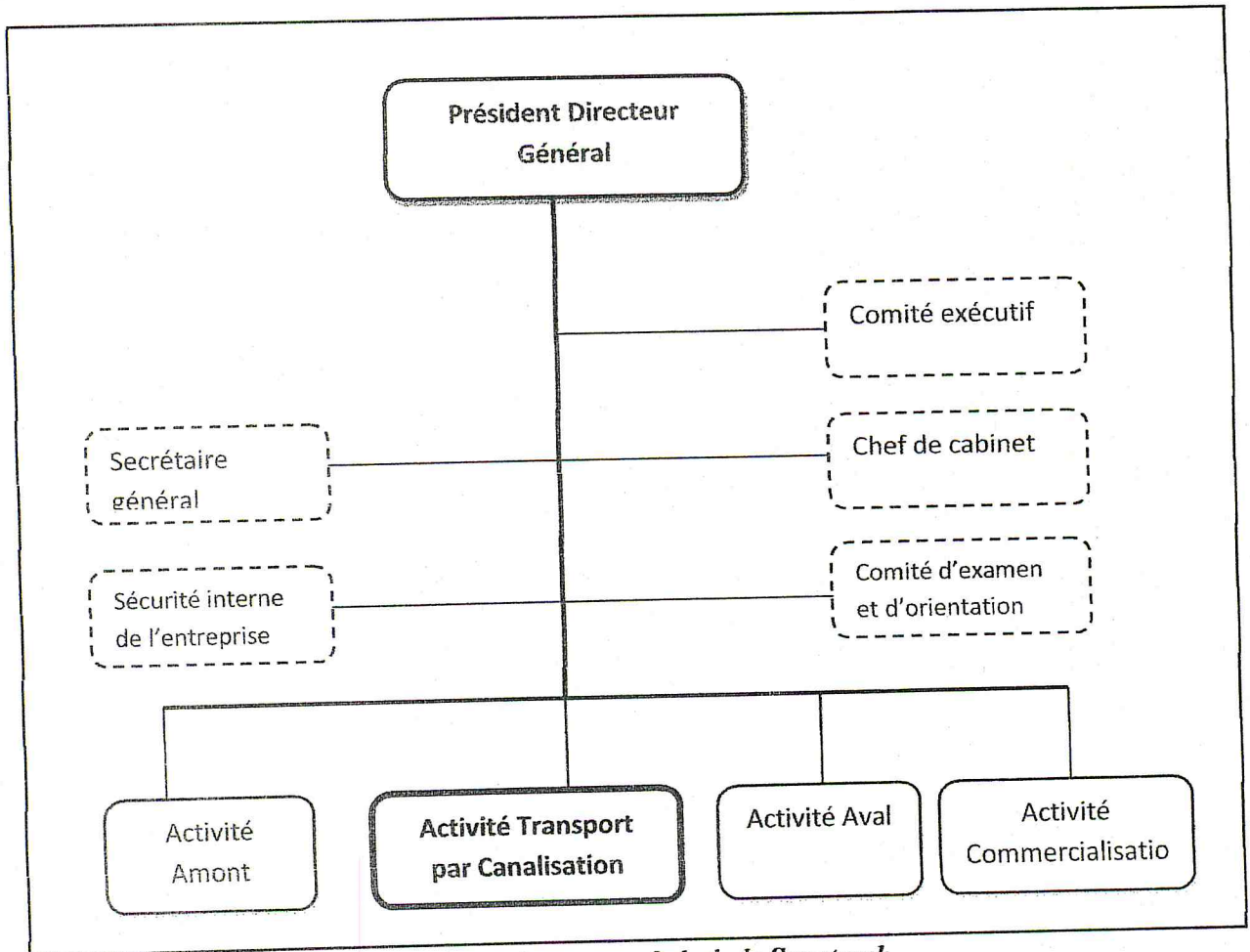
Pour conclure, nous pouvons dire, qu'en apportant une amélioration au service, nous ne pourrions qu'être satisfaites de notre travail. Nous avons la certitude d'avoir éclairci de nombreux doutes et d'avoir comblé en grande partie les lacunes des utilisateurs.

Et Avec les résultats auxquels nous sommes arrivés, l'objectif a été atteint. Néanmoins ce travail reste toujours sujet à nouvelle version.

*Nous espérons que la lecture de ce mémoire a été claire et agréable*

## Annexe A : Contexte de travail

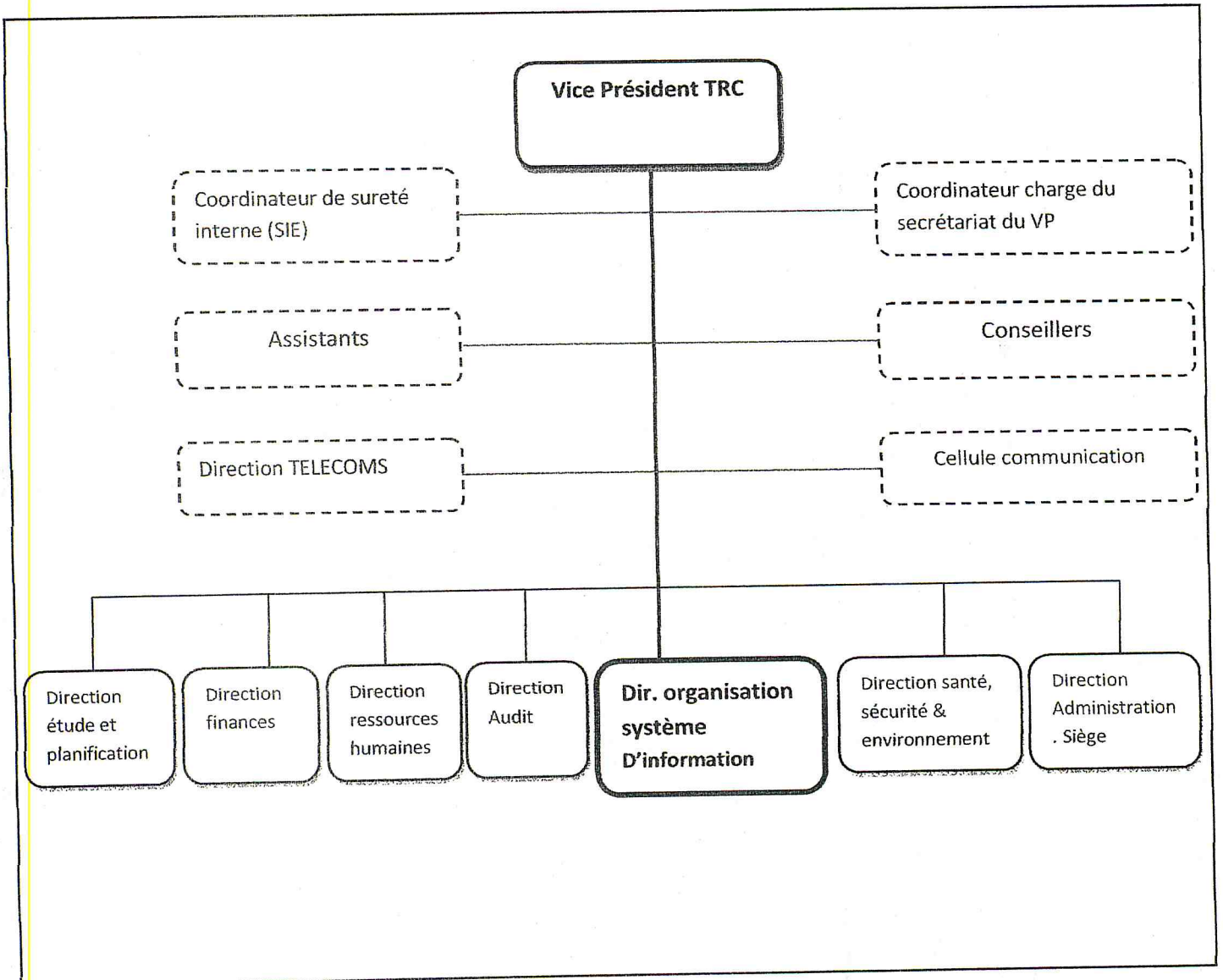
Sonatrach est la compagnie algérienne de recherche et d'exploitation, de transport et de transformation et la commercialisation des hydrocarbures et de leurs dérivés.



*Figure36 : l'organigramme général de Sonatrach*

### 1. Activité Transport par Canalisation(TRC)

Et Au sein du groupe Sonatrach, l'Activité Transport par Canalisation TRC a la charge d'acheminer les hydrocarbures, de définir, de réaliser, d'exploiter et d'assurer la maintenance en coopérant avec le marché national et internationale.



*Figure37 : Partie de l'organigramme général de l'activité TRC/Sonatrach*

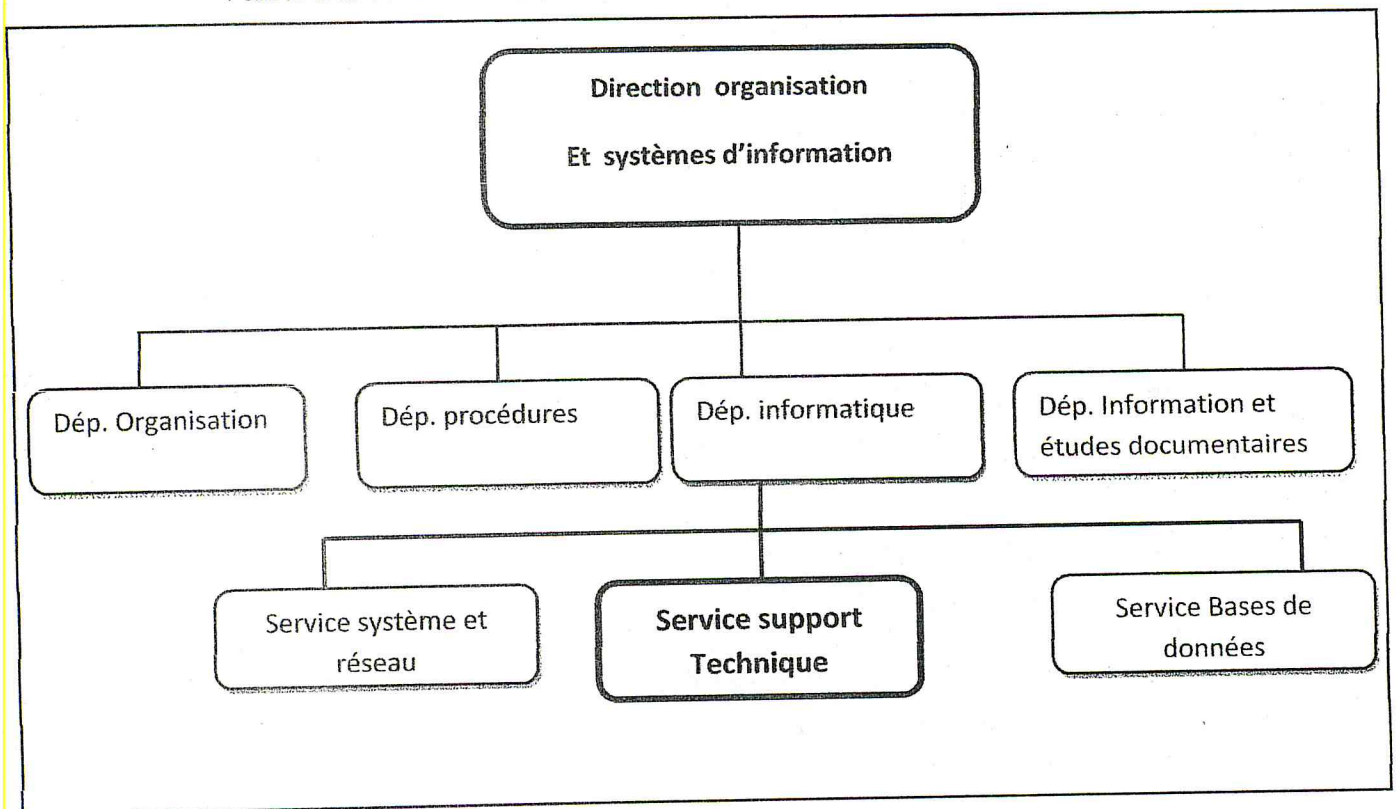
## 2. La structure d'accueil : Direction organisation system d'information

La Direction organisation et system d'information est chargée d'identifier et d'assembler tout les besoin informatique de l'activité TRC.



### 2.1. Ses missions

- ✓ Mettre en œuvre la politique informatique du group Sonatrach à TRC
- ✓ Elaborer le plan directeur informatique de l'activité
- ✓ Veiller à la disponibilité, l'administration, la sécurité et le développement des composantes techniques du système d'information. Ces composantes sont :
  - L'infrastructure réseau Lan et Wan.
  - Les systèmes informatiques.
  - Les services réseau (Internet, Intranet et messagerie).
  - Les applications et les bases de données.
- ✓ Veiller à la coordination des travaux des structures informatiques de l'activité.



*Figure38 : Organisation de la structure d'accueil activité TRC/Sonatrach*



## **Annexe B: RUP (Rational Unified Process) :**

Parmi les processus de développement, RUP tient une place à part. en premier lieu parce que ses concepteurs sont des légendes de l'objet :Ivar Jacobson, Grady Booch et James Rumbaugh. Ensuite parce qu'il s'agit d'un produit. En effet, RUP est l'unique implémentation commerciale d'UP (Unfied Process), un processus théoriquement « standard » élaboré par le même trio.

### **1. Les 3 fondements :**

RUP est cité par ses auteurs :

« Piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental. »

Si cette phrase n'a pas la même portée que les mantras d'XP, elle constitue cependant une description synthétique des fondements de RUP (et d'UP)

#### **1.1. Piloté par les cas d'utilisation**

Les cas d'utilisation, dans RUP, ne servent pas qu'à améliorer la capture des exigences fonctionnelles des utilisateurs. Ils servent d'intrant aux différentes activités du processus : analyse, conception, implémentation, rédaction des tests et même déploiement. Le projet dans son ensemble a pour objectif de réaliser les cas d'utilisation. Ces derniers sont donc :

- ✓ Détaillés (puis spécifiés en terme informatique) par l'analyse ;
- ✓ Réalisés par la conception.
- ✓ Concrètement implémentés par les développeurs (sous la forme de modèles d'implémentation et de code source).
- ✓ Vérifiés par les scénarios de test.

## 1.2. Centré sur l'architecture

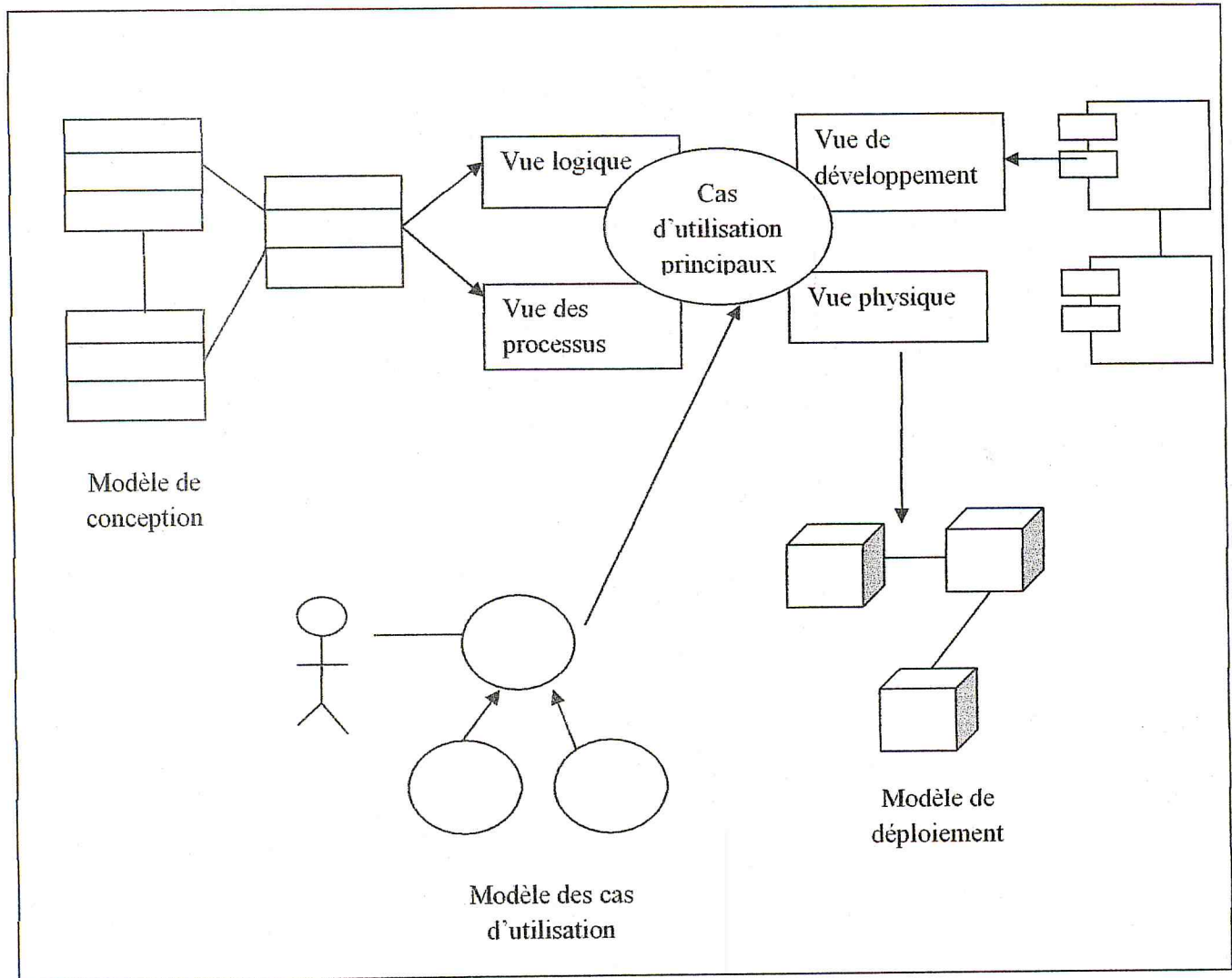
RUP est « centré sur l'architecture » simplement parce que l'élaboration de celle-ci est une tâche faisant partie intégrante du processus (ce qui n'est pas toujours le cas). RUP reconnaît que l'architecture est un compromis entre les besoins des utilisateurs, les contraintes techniques des plates-formes de production (système d'exploitation, middleware, etc.) et les ressources (humaines, techniques ...) disponibles. Une application implémente les exigences des utilisateurs mais sa réalisation ne peut faire abstraction du contexte, en particulier de l'existant.

La description d'une architecture n'est pas monolithique, elle est en fait constituée de l'agrégation cohérente de différents points de vue : celui de l'analyste métier, celui du développeur, celui de l'ingénieur système, etc. RUP préconise d'utiliser le modèle des 4+1 vues pour guider l'élaboration de l'architecture.

Ce modèle comprend quatre vues principales et une vue « coordinatrice » :

- ✓ La vue logique qui correspond à la structure de l'application en termes de classes, de sous-systèmes, de couches, etc.
- ✓ La vue des processus, qui correspond à une description des tâches exécutées par l'application (processus, threads), et de leurs relations ;
- ✓ La vue de développement qui est une traduction concrète du modèle logique (structurée en package, en bibliothèques, etc) et qui est utilisée pour l'allocation des tâches aux développeurs.
- ✓ La vue physique qui constitue une description de la répartition de l'application dans les différents nœuds (machines, serveurs) de la plate-forme de production.
- ✓ La vue des cas d'utilisation qui sert à cadrer les vues précédentes.

Le rôle de cette dernière vue (la plus unifiée) s'explique par la manière dont RUP propose de gérer la réalisation de l'architecture.



*Figure 39 : Modèle 4+1 vues*

### 1.3. Itératif et incrémental

RUP, comme la plupart des nouveaux processus de développement est itératif. L'objectif recherché est toujours le même : favoriser une diminution progressive des risques dès le début du projet. Il s'agit de se prémunir contre des défauts les plus douloureux du modèle en cascade dans lequel les risques commencent à diminuer qu'en phase d'intégration. C'est-à-dire beaucoup tardivement.

RUP est aussi incrémental car chaque itération est planifiée. Chaque itération est un projet à l'intérieur du projet qui met à jour une partition de l'ensemble des artefacts. Cependant les itérations de RUP ne sont pas toujours identiques. Les itérations s'inscrivent dans quatre phases successives.



Les itérations de RUP sont donc des spécialisations d'un modèle d'itération générique qui consiste dans l'enchaînement des activités classique : récolte des exigences, analyse, conception, implémentation et test. Au fil de l'avancement du projet, les poids respectifs des différentes activités vont évoluer.

## **2. Les disciplines et leurs relations**

Les disciplines sont des groupes d'activités qui se répètent théoriquement à chaque itération. En pratique, la charge allouée à une discipline donnée va varier en fonction de l'avancement du projet. En début de projet, l'accent est mis sur la capture des exigences alors qu'en fin de projet, l'essentiel du travail est consacré à la réalisation de l'application. Cependant, à l'inverse d'un processus de développement en cascade, RUP insiste sur la nécessité de ne pas cantonner à une phase du projet. RUP est tout simplement itératif

### **2.1. Modélisation du métier (Business modeling)**

Cette discipline vise à produire un modèle de l'organisation à laquelle est destinée l'application. Il s'agit d'identifier les acteurs, les processus métiers, la vision métier que le projet doit implémenter.

La charge allouée à cette discipline varie selon le type et le contexte de réalisation de l'application : nouveau métier dans l'entreprise, refonte d'une application existante, automatisation de processus manuels, etc. si les processus métier sont parfaitement maîtrisés par l'équipe de développement, cette discipline peut être ignorée.

### **2.2. Analyse et conception**

L'objectif des activités de cette discipline est de traduire les cas d'utilisation en artefact de conception du système. Cette réalisation des cas d'utilisation doit aussi tenir compte de l'environnement d'implémentation et pilote la conception.



### 2.3. Implémentation

L'implémentation vise :

- ✓ A raffiner le modèle de conception pour diminuer la granularité de l'organisation du code de l'application (en définissant des composants) ;
- ✓ A écrire effectivement le code source et les tests unitaires associés
- ✓ A intégrer le travail d'implémentation produit séparément par plusieurs implémenters.

### 2.4. Déploiement

Cette discipline prend en charge les activités liées au conditionnement du produit en vue de sa mise à disposition des utilisateurs. Cette discipline n'intervient pas seulement en fin de projet. Elle est aussi mise en œuvre dès les premières itérations du projet pour produire des versions préliminaires du produit. Ces versions, selon leur degré d'avancement, resteront privées (pour l'équipe du projet) ou seront distribuées à un panel d'utilisateurs en vue de leur évaluation.

### 2.5. Test

Les tests ont un rôle central dans RUP en raison notamment de la bonne pratique « vérifier continuellement la qualité ». Cette discipline n'est pas mise en œuvre uniquement avant ou après le déploiement mais continuellement pendant les développements. Dans un processus itératif comme RUP qui, de plus, favorise la « parallélisation » des développements, la réalisation de tests de non-régression est une activité fondamentale. Il s'agit de vérifier, par l'intermédiaire de tests unitaires (étroitement associés au code, à un niveau de granularité faible) et de tests fonctionnels, que les cas d'utilisation implémentés dans les premières itérations continuent à fonctionner dans les suivantes.

## 3. Les phases du RUP

RUP est découpée en quatre phases qui se terminent lorsque leurs objectifs (exprimés en artefacts) sont validés.

### 3.1. Inception

La charge allouée à cette phase varie selon qu'il s'agit du développement d'un nouveau produit ou de la maintenance d'un produit existant. Dans ce dernier cas, elle peut être notablement réduite. L'objectif de l'inception est de trouver un compromis entre les exigences et les contraintes (financières, délais, ressources, etc.) avancées par les différents stakeholders. Il s'agit de s'assurer que les buts du projet sont compris et acceptés. Cette phase doit ensuite se terminer sur une première version réaliste du planning s'appuyant sur un nombre réduit de cas d'utilisation importants.

### 3.2. Elaboration

L'objectif principal de l'élaboration est la stabilisation de l'architecture de référence, l'élimination ou l'identification des risques majeurs. Cette démarche sera validée par la réalisation d'une documentation de l'architecture de référence, éventuellement illustré par différents prototypes, et une nouvelle version, plus réaliste, du planning.

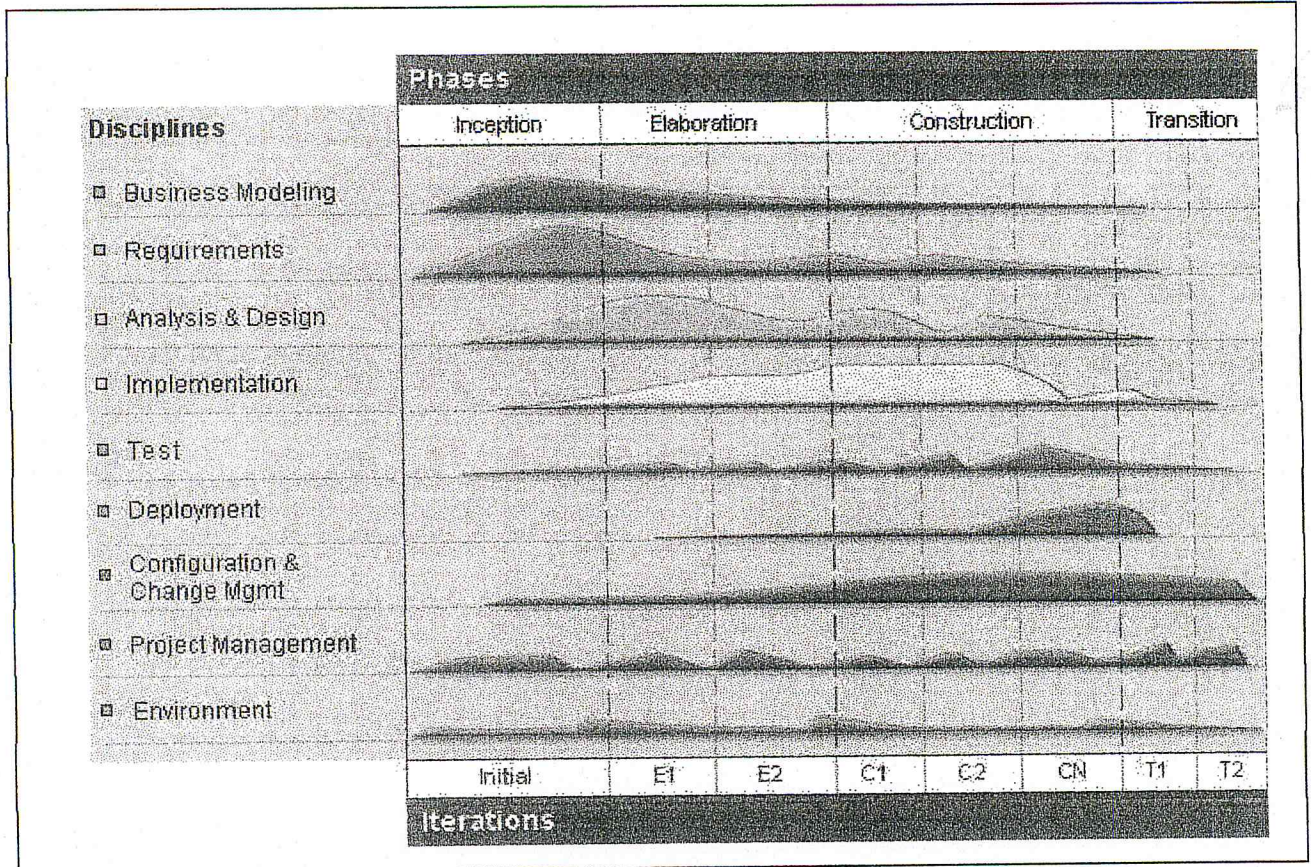
### 3.3. Construction

La construction est marquée par une forte augmentation de la charge affectée à la discipline implémentation. Elle vise à finaliser la capture des exigences et à produire successivement des versions exécutables de l'application de plus en plus proches de la cible.

### 3.4. Transition

Cette version vise à consolider le produit pour en faire une version finalisée. Elle se termine par la transmission des artefacts du projet à un tiers. La complexité de cette phase varie selon les cas. Elle dépend de ce que le tiers aura comme responsabilité vis-à-vis de l'application (production, maintenance corrective, maintenance évolutive, intégration ...) et donc de la quantité de connaissances à opérer. **[RUP]**

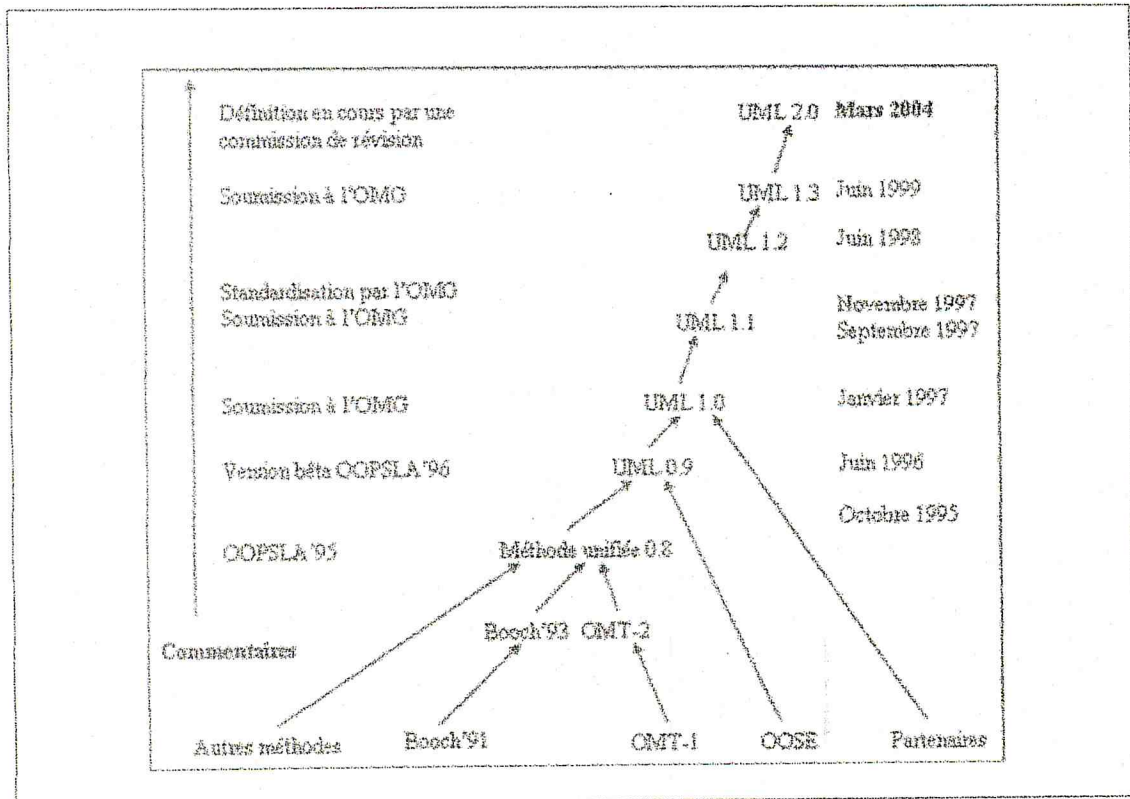




*Figure 40 : Les deux dimensions de RUP*

## Annexe C : UML (Unified Modeling Language):

UML (Unified Modeling Language) traduit "langage de modélisation objet unifié", est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 ; OMT, Booch et OOSE.



**Figure 41 : l'évolution de la norme UML**

- UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue, permet de définir et de visualiser un modèle, à l'aide de diagrammes.

### 1. Les diagrammes d'UML :

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle, UML 2 s'articule autour de 13 diagrammes différents, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel.

Par ailleurs, Les 13 diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.



### 1.1. Diagrammes structurels ou statiques :

- ◆ **Diagramme de classes:** il représente les classes intervenant dans le système.
- ◆ **Diagramme d'objets :** il sert à représenter les instances de classes (objets) utilisées dans le système.
- ◆ **Diagramme de composants:** il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...etc.).
- ◆ **Diagramme de déploiement:** il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
- ◆ **Diagramme des paquetages:** un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le diagramme de paquetage sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions.
- ◆ **Diagramme de structure composite :** permet de décrire sous forme de boîte blanche les relations entre composants d'une classe.

### 1.2. Diagrammes comportementaux :

- ◆ **Diagramme des cas d'utilisation :** il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
- ◆ **Diagramme états-transitions:** permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.
- ◆ **Diagramme d'activité:** permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

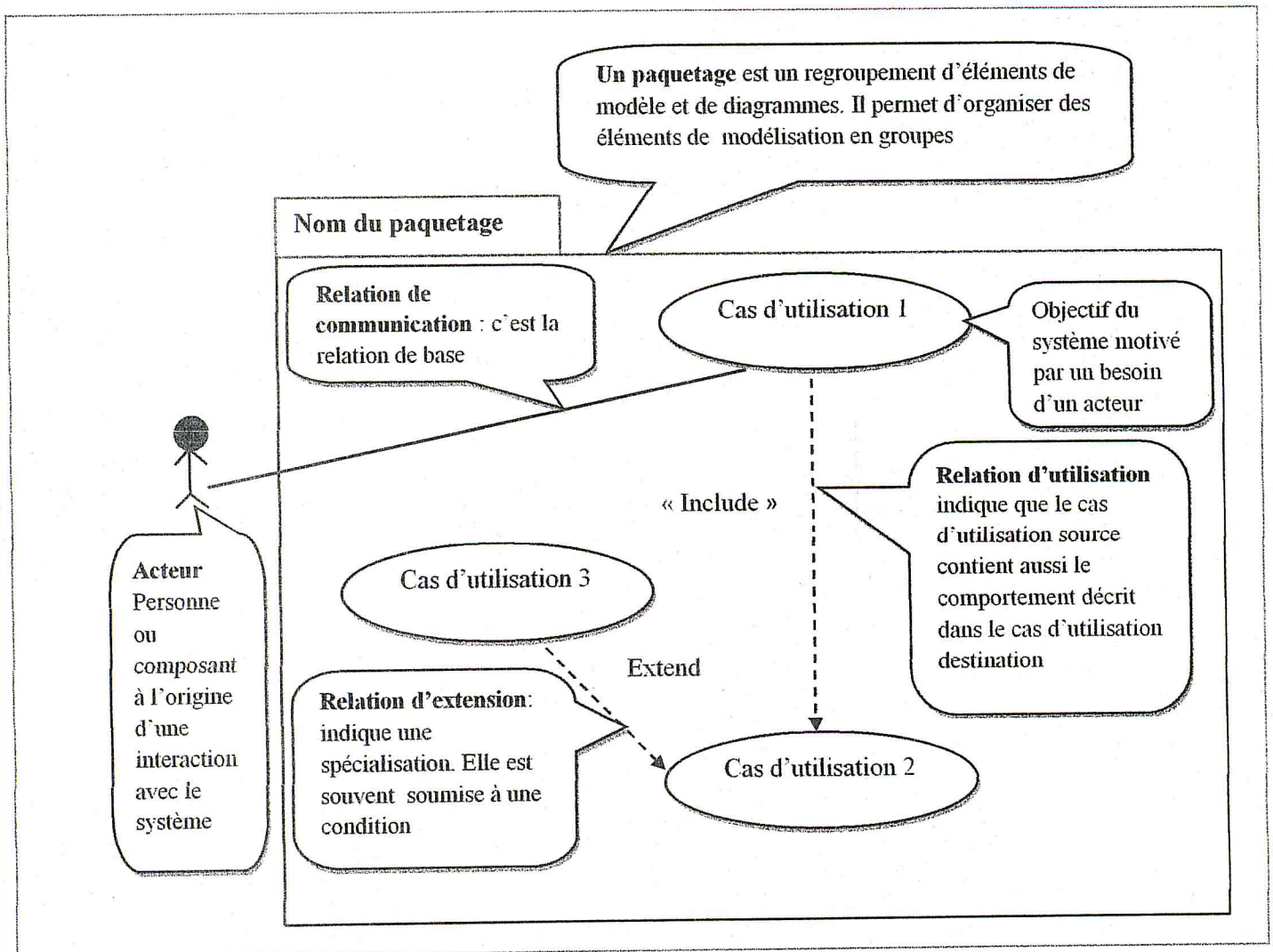
### 1.3. Diagrammes d'interaction ou dynamiques :

- ✦ **Diagramme de séquence:** représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
- ✦ **Diagramme de communication:** représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- ✦ **Diagramme global d'interaction:** permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).
- ✦ **Diagramme de temps:** permet de décrire les variations d'une donnée au cours du temps.

## 2. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation décrit sous la forme d'action et de réaction le comportement d'un système de point de vue de l'utilisateur et donc le caractère fonctionnel des objets.

### 2.1. Formalismes graphiques et définitions :



**Figure42 : formalisme graphique du diagramme de cas d'utilisation**

### 3. Diagramme de classes :

Un diagramme de classes montre la structure statique d'un système. Il permet la visualisation des classes et d'expliquer ce qu'il faut réaliser plutôt que d'expliquer comment le réaliser.

#### 3.1. Formalisme graphique et définitions :



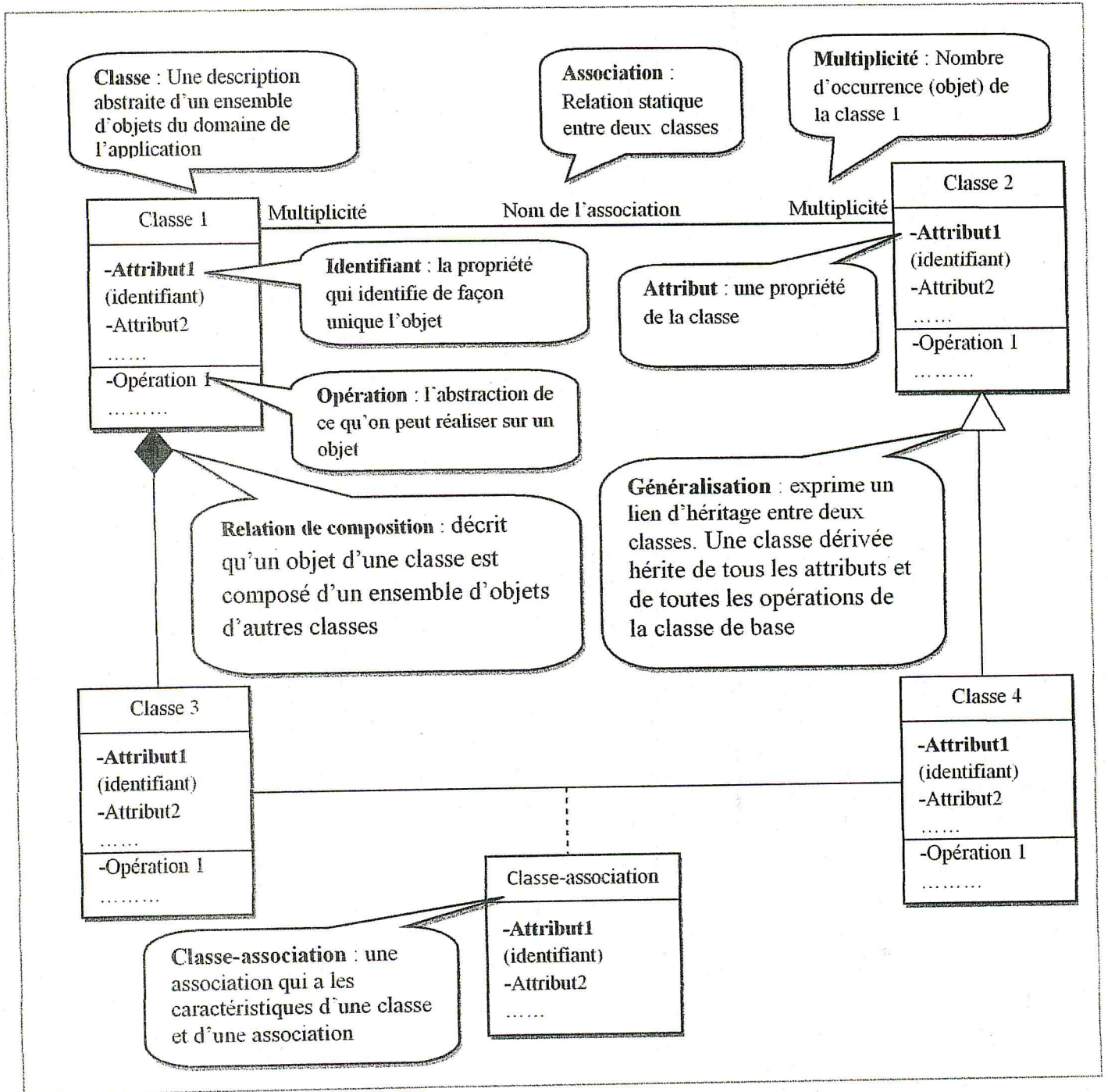


Figure43 : formalisme graphique du diagramme de classes

#### 4. Diagramme de séquence :

Un diagramme de séquence est une représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.

##### 4.1. Formalisme graphique et définitions :

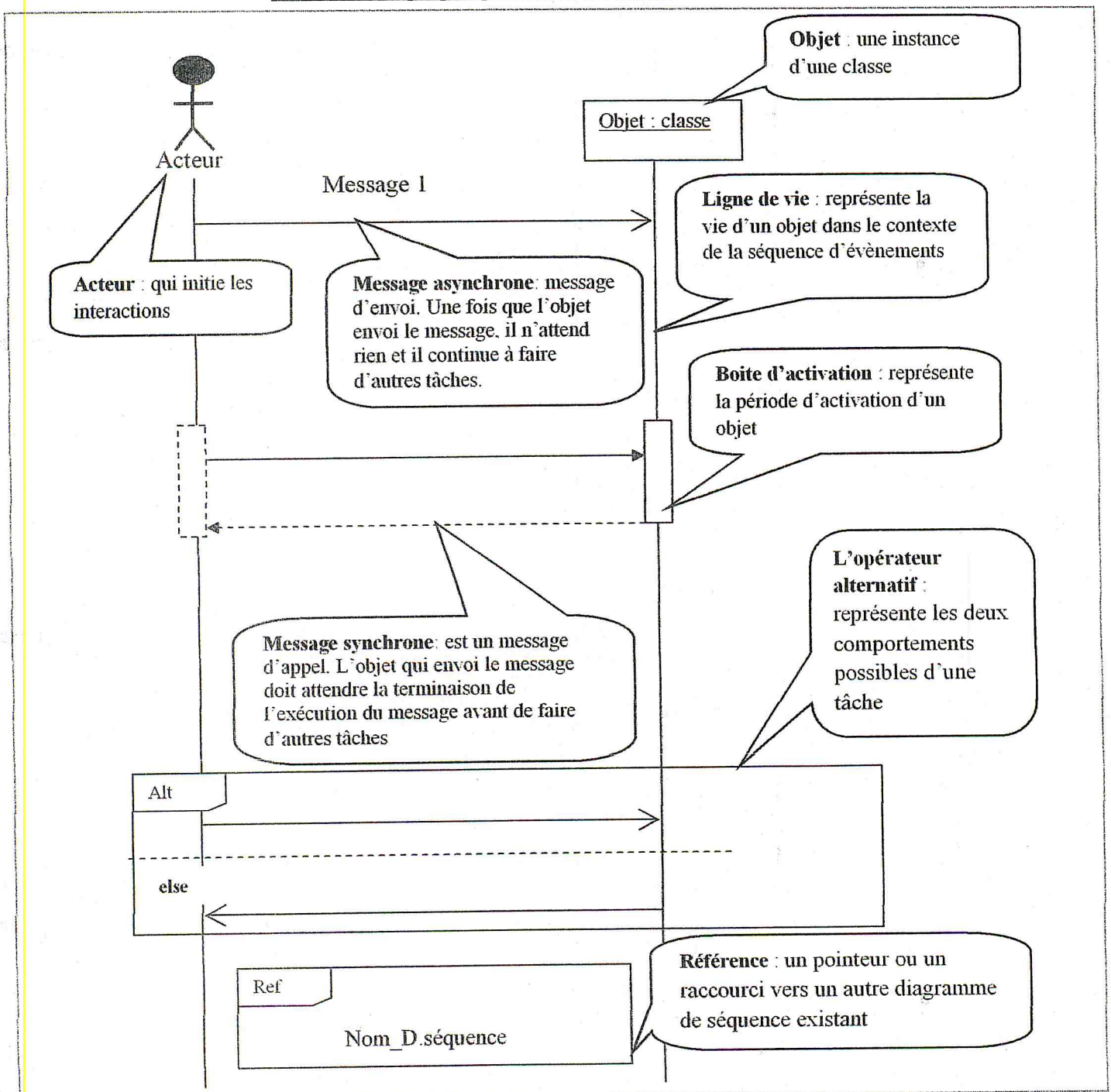
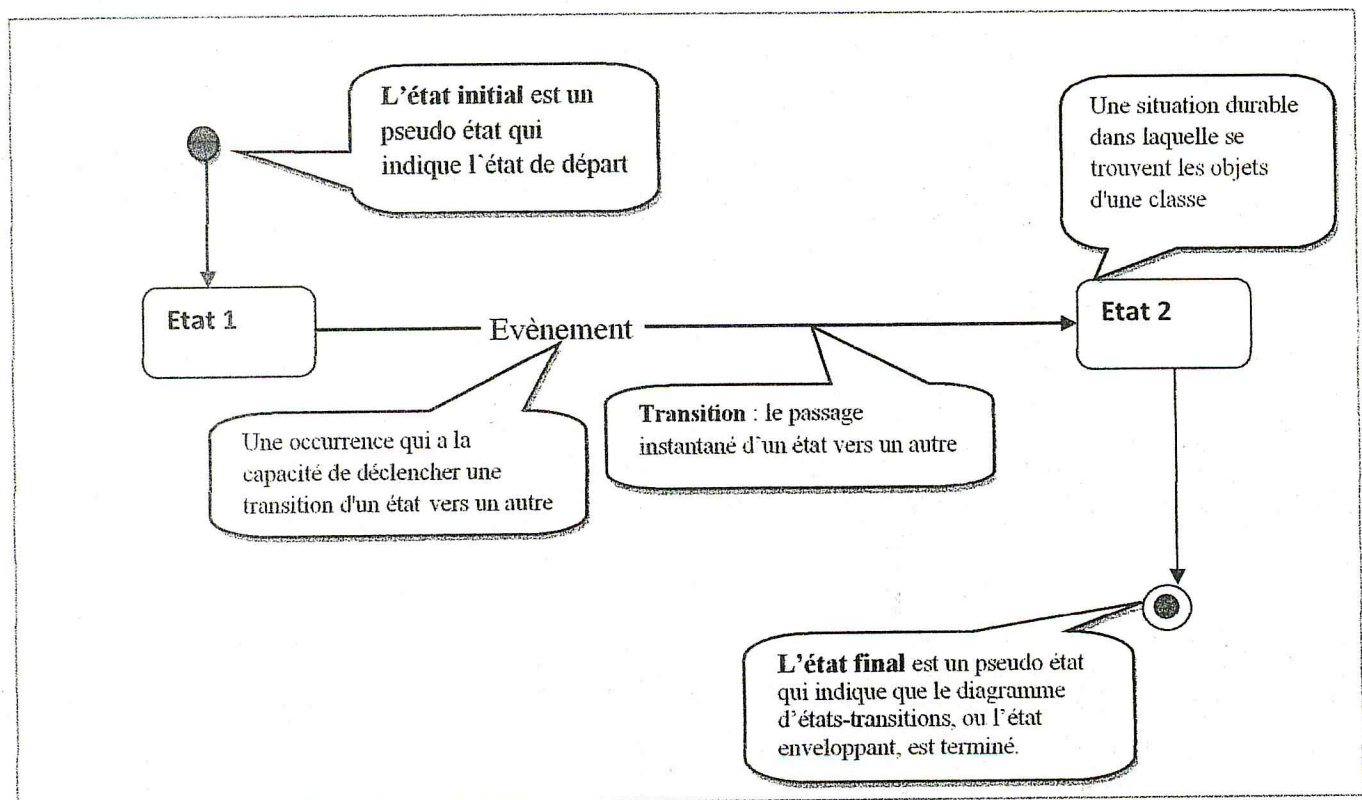


Figure44 : formalisme graphique du diagramme de séquences

## 5. Diagramme d'état-transition

Les diagrammes d'état-transition sont des automates à états. Composés d'états, de transitions, d'événements et d'activités. Ils présentent la vue dynamique d'un système. Le diagramme d'états-transition est associé à une classe pour laquelle on gère différents états : il permet de représenter tous les états possibles ainsi que les événements qui provoquent les changements d'état.

### 5.1. Formalisme graphique et définitions :



*Figure45 : formalisme graphique du diagramme d'états-transition*



## 6. Les points forts d'UML :

- *UML est un langage formel et normalisé*
  - ✓ Gain de précision.
  - ✓ Gage de stabilité.
  - ✓ Encourage l'utilisation d'outils.
- *UML est un support de communication performant*
  - ✓ Il cadre l'analyse.
  - ✓ Il facilite la compréhension de représentations abstraites complexes.
  - ✓ Son caractère polyvalent et sa souplesse en font un langage universel.

## 7. Les points faibles d'UML :

- *La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.*

La nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

- *Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.*

Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.

Les auteurs d'UML sont tout à fait conscients de l'importance du processus, Mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard. [UML\_action]

## Annexe D : Oracle

### 1. Oracle aujourd'hui

La société Oracle System Corporation, basée à Redwood Shores en Californie, conçoit des produits logiciels et offre des services de gestion électronique des informations. Sa renommée est mondiale et ses revenus excèdent les huit milliards de dollars. Elle est implantée dans plus de 140 pays, et ses logiciels sont utilisés sur plus de cent systèmes différents. Elle occupe aujourd'hui une place prépondérante dans le monde de l'information.

Oracle joue un rôle important dans l'adoption du modèle informatique d'Internet, dans lequel les organisations s'orientent vers le web pour y déployer des applications distribuées. Etant donné la mouvance vers une approche basée composants, Internet associé à l'architecture client-serveur trois tiers représente l'infrastructure idéale pour la fourniture d'applications distribuées. Les trois tiers ou niveaux qui composent cette architecture sont les suivants :

- ✓ Le client : parfois appelé niveau présentation, sur lequel les informations sont saisies par l'utilisateur avant d'être traitées ou sur lequel sont affichées des données renvoyées en réponse à une requête.
- ✓ Le serveur d'applications : sur lequel les traitements sont effectués en fonction de la logique implémentés par le système. Les règles de gestion en vigueur y sont appliquées, l'intégrité des données est contrôlée, et les traitements complexes liés aux exigences du système sont exécutés. Ce niveau est la clé de voûte de l'architecture trois tiers.
- ✓ Le serveur de base de données : situé en arrière-plan, sur lequel interviennent les opérations de stockage des données et les opérations de recherche d'informations en réponse aux requêtes des deux autres niveaux. Dans de nombreuses situations, il s'agit d'une base de données relationnelle spécialement optimisée pour gérer les informations en provenance ou en direction du serveur d'applications lorsque les utilisateurs interagissent avec le système.

## **2. Serveur de base de données (SQL\*Plus)**

SQL\*Plus est un système de gestion de base de données relationnel (SGBDR) fourni par Oracle Corporation. Il a été développé par Lawrence Ellison.

Oracle est écrit en langage C, il est disponible sur de nombreuses plates-formes matérielles.

### **2.1. Les fonctionnalités:**

- ✦ La définition et la manipulation des données.
- ✦ La cohérence des données.
- ✦ La confidentialité des données.
- ✦ L'intégrité des données.
- ✦ La sauvegarde et la restauration des données.
- ✦ La gestion des accès concurrents.

## **3. Oracle Developer**

Avec les déploiements actuels sur des plates-formes client-serveur, Internet-Intranet, la version 6 est déjà connue comme outil de développement robuste. Il existe sur le marché un large éventail de produits de développement basés sur une interface graphique ; Form Builder est l'offre d'Oracle dans ce domaine.

## **4. Que représente Form Builder ?**

Forms Builder version 6 est un composant très complexe de Developer, la nouvelle génération d'outils Oracle permettant le développement d'applications pour le web.

Avec cet ensemble intégré, les programmeurs peuvent concevoir des programmes pour un intranet d'entreprise ou pour internet tout en continuant à travailler avec les implémentations client-serveur existantes.

Form Builder est un outil riche en fonctionnalités qui peut produire des écrans professionnels de qualité en utilisant des données stockées dans une base. En utilisant la technologie OLE2, il peut incorporer des graphiques, du son, de la vidéo, des



documents de traitement de texte et des tableaux. Les autres éléments qui peuvent être liés, nouveaux dans la version 6, sont les composants JavaBeans et ActiveX.

Oracle a fait beaucoup d'efforts pour garantir que les objets déjà développés, en interne ou par une société tierce, puissent continuer à être utilisés et réutilisés dans Form Builder. L'accent a aussi été mis sur la productivité des programmeurs avec Developer. En effet, Forms Builder bénéficie d'une interface pouvant être principalement guidée au moyen de la souris et offre la capacité de développer sans trop de codage.

### **5. Serveur d'application et Oracle Developer**

Le développement d'applications évolue actuellement du modèle client-serveur, connu sous la désignation de modèle deux tiers, vers une architecture multi-tiers (souvent appelée trois-tiers ou n-tiers). Un tiers est tout simplement un niveau de composant logique.

Oracle possède son propre serveur d'applications supportant l'architecture multi-tiers, qui rassemble les avantages des modèles client-serveur et Internet dans une seule application. Dans une application web, la logique applicative et les règles de gestion sont concentrées sur le tiers central, le serveur d'application, au lieu d'être mise en application sur les postes des utilisateurs. [Ora2]

### **6. Avantages**

- ✓ Richesse fonctionnelle
- ✓ Fonction d'audit évolué
- ✓ Parallélisme; haute disponibilité; grande possibilité de tuning
- ✓ Procédures stockés en PL-Sql (langage propriétaire Oracle, orienté ADA) ce qui peut s'avérer utile pour les équipes de développement.
- ✓ Assistants performants via Oracle Manager Server, possibilité de gérer en interne des tâches et des alarmes
- ✓ Gestion centralisée de plusieurs instances

- ✓ Pérennité de l'éditeur : avec plus de 40% de part de marché, ce n'est pas demain qu'Oracle disparaîtra
- ✓ Accès aux données système via des vues, bien plus aisément manipulable que des procédures stockées.
- ✓ Interface utilisateur remaniée et extrêmement riche
- ✓ Services Web, support XML
- ✓ Ordonnanceur intégré
- ✓ Compression des données et des sauvegardes
- ✓ Support technique Metalink extrêmement riche [*developpez.com*]

## **7. Inconvénients Oracle :**

- ⇒ Prix exorbitant, tant au point de vue des licences que des composants matériels (RAM, CPU) à fournir pour de bonnes performances.
- ⇒ Administration complexe liée à la richesse fonctionnelle.
- ⇒ Porosité entre les schémas ce qui implique la difficulté de faire cohabiter de nombreuses applications sans devoir créer plusieurs instances.
- ⇒ Une quantité des bugs proportionnels à la richesse fonctionnelle, surtout sur les dernières versions.
- ⇒ Nombreuses failles de sécurités liées à l'architecture elle-même. [*Oracle.com*]

## **8. Les exigences de l'implémentation**

### **8.1. Configuration du client**

Avant que les communications puissent être mises en œuvre, le poste client doit être physiquement connecté au réseau et le logiciel de communication doit être installé. La carte réseau du client doit être supportée par le logiciel de communication.

### 8.1.1. Caractéristiques matérielle et logicielles du client

Le poste client doit satisfaire aux exigences suivantes :

Matériel : IBM, Compaq ou PC compatible à 100% avec un processeur 80286 ou supérieur

Suffisamment d'espace disque pour stocker les fichiers sur le système d'exploitation, Net8.

Une unité de disque à utiliser durant l'installation.

Une carte réseau pour les communications réseau.

Mémoire :

Suffisamment de mémoire pour exécuter le logiciel de réseau, Net8, et votre application, avec un minimum de 16Mo.

Logiciel :

Microsoft windows;

net8 pour Windows avec l'adaptateur TCP/IP ;

Logiciel de communication.

Le logiciel TCP/IP de net8 pour windows requiert 100Ko d'espace disque et environ 120Ko de mémoire



## *Bibliographie*

---



**[Arch1]** : Mounir GRARI. Mémoire Principes et états de l'art de l'approche MDA et applications pour des plates-formes PHP orienté 3-tiers.

**[Arch2]** : Rémi LEBLOND. Vers une architecture n-tiers.

<http://www.metz.supelec.fr/metz/personnel/vialle/course/SI/n-tiers/remi.leblond.free.fr/probatoire.pdf> (2003).

**[RUP]** : Pierre Yves Cloux. RUP-XP : Architecture et outils (industrialiser le processus de développement). Dunod, Paris (2003)

**[UML2]** : Benoit Charroux, Aomar Osmani, Yann Thierry Mieg. UML2. Pearson Education (2005).

**[UML\_action]** : Roques.Pascal. UML En Action : De l'analyse des besoins à la conception J2EE. Eyrolles (2006).

**[Proc]** : Pierre Gérard. Processus de Développement Logiciel (Université de Paris 13).  
<http://www-lipn.univ-paris13.fr/~gerard/images/user/cours/methodo-support.pdf> (2008)

**[Ora1]** : Kevin Loney et Marlene Theriault. Oracle 8 i DBA : configuration, administration, sécurité, architecture client/serveur. Oracle Press (2000).

**[Ora2]** : Michael Abbey et Michael J. Corey et IAN Abramson. Oracle 8i Notions fondamentales. Oracle Press(1999).



## *Sites utiles*

---

[www.developpez.com](http://www.developpez.com)

[www.oracle.com](http://www.oracle.com)

[www.araboug.org](http://www.araboug.org)