الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE Saâd DAHLAB de BLIDA

FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du diplôme d'Ingénieur d'Etat En Génie Informatique

Option Système d'Information

Présenté par :

ATTOU Sofiane
HADJAR Lies

THEME

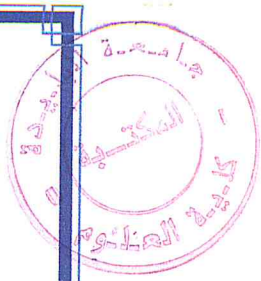
REALISATION D'UN LOGICIEL POUR LA GESTION ET LA LOCALISATION GEOGRAPHIQUE DES BUREAUX DU SIEGE SONATRACH

Suivi par : M^elle^ N.BOUSTIA

Encadré par : Mr H.HOCINE

Promotion 2003 – 2004

Organisme d'accueil : SONATRACH



بسم الله الرحمن الرحيم

اللهم لا تدعني أصاب بالغرور إذا نجحت،

و لا أصاب باليأس إذا فشلت

بل علمني دائما بأن الفشل هو التجربة التي تسبق النجاح

اللهم علمني أن التسامح هو أكبر مراتب القوة،

و أن حب الانتقام هو أول مظا هر الانتقام

اللهم إذا جردتني من المال اترك لي الأمل،

وإذا جردتني من النجاح اترك لي قوة الصبر حتى أتغلب على الفشل،

وإذا جردتني من نعمة الصبر اترك لي نعمة الإيمان

اللهم إذا أسأ ت إلى الناس اعطيني شجاعة الاعتذار

وإذا أساء لي الناس اعطيني شجاعة العفو.

Remerciments

Nous remercions d'abord ALLAH le tout puissant de nous avoir donner la force, la patience et la volonté pour achever ce travail.

Nous exprimons ici notre vive reconnaissance au personnel de la SONATRACH, Nous pensons particulièrement à :

Monsieur HOCINE Hakim, notre encadreur, pour nous avoir encadrer durant la réalisation de ce travail, pour son aide, sa disponibilité et ses orientations tout au long de notre stage.

Mme HADDAD, Mme SEDDIK, Mme BENKHELFALLAH, Mme KHEDECH, Mlle SEDDIK, Mme KABLA pour leurs présences aide et leurs conseils.

Mr MORSLI, Mr ALOUI, Mr YAHYAOUI, Mr BELKHIER, Mr SAHRAOUI, Mr SAADI, Mr HAFFAF, Mr MITICHE, Mr CHIOUKH, Mr OULD KADDOUR.

Mr DREBINE Mourad pour son soutien et aide.

Toute l'équipe de SPE et CTI.

Nous tenons à remercier également les enseignants de l'Université de BLIDA en particulier :

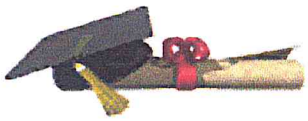
Madame BENSETTITI Souad, le chef de département informatique, dont l'appui, la rigueur, la détermination, les orientations et la disponibilité malgré ses occupations ont contribué à mener à bien ce projet, cette formation et ont fait que ce jour soit la sortie de la deuxième promotion des informaticiens ingénieurs de l'USDB.

Mademoiselle BOUSTIA Narhimene qui a suivi notre travail durant toute cette année de stage, et qui nous a prodigué aussi bien ses connaissances que son soutien.

Nous remercions également l'ensemble des enseignants de l'USDB qui nous ont encadré au cour de notre cursus et a qui revient en vérité le fruit de ce labeur surtout : Mr HADJ YAHIA, Mr KOUDIL, Mme OUAHRANI HARKATI, Mme ABED.

Mlle AOUSSAT pour son soutien.

Que tous ceux qui ont contribué de prés ou de loin à la réalisation de ce modeste travail, trouvent ici notre sincère reconnaissance.



Dédicaces

Je tiens à dédier ce modeste mémoire :

A ma très chère **M**ère et à mon cher **P**ère, en témoignage et en gratitude de leurs dévouement, de leurs soutien permanent durant toutes mes années d'études, leurs sacrifices illimités, leurs réconfort moral, eux qui ont consenti tant d'effort pour mon éducation, mon instruction et pour me voir atteindre ce but, pour tout cela et pour ce qui ne peut être dit, mes affectations sans limite.

A ceux qui sont la source de mon inspiration et mon courage, à qui je dois de l'amour et de la reconnaissance :

A mes deux chères **G**rands-**M**ères. (Que Dieu les gardes pour moi)

A mon **C**her **F**rère Yacine et a mes **C**hères **S**œurs : Amel et Nadia.

A mon **O**ncle Mohamed, sa **F**emme Fadhila et leurs deux **F**illes : Yasmine et Houda.

A toute ma famille.

A mon cher **A**mi et **B**inôme Lies pour tous les moments de joies et de peines qu'on a passé ensemble, A sa Famille aussi.

A mes Voisins et a mes Amis (es) surtout ceux qui ont supporté mes sauts d'humeur.

A mon **P**romoteur BOUSTIA Narhimene : les mots ne suffisent pas pour vous exprimer ma profonde gratitude et mes vifs remerciements pour tous ce que vous avez fait pour moi.

A la mémoire de :

> La famille KADDI, puisse Dieu le tout puissant accorder aux disparus du séisme sa sainte miséricorde et les accueillir en son vaste paradis.

A ceux qui ont cru en moi,

A ceux qui croient en moi,

et A ceux qui croiront toujours en moi.

A vous tous un grand merci.

Sofiane

Dédicaces

Je tiens à dédier ce modeste mémoire :

A ma très chère Mère et à mon cher Père, en témoignage et en gratitude de leurs dévouement, de leurs soutien permanent durant toutes mes années d'études, leurs sacrifices illimités, leurs réconfort moral, eux qui ont consenti tant d'effort pour mon éducation, mon instruction et pour me voir atteindre ce but, pour tout cela et pour ce qui ne peut être dit, mes affectations sans limite.

A ceux qui sont la source de mon inspiration et mon courage, à qui je dois de l'amour et de la reconnaissance :

A la mémoire de mon Grand-Père.

A ma chère et tendre Grand-Mère. (Que Dieu la garde pour moi)

A mon Cher Frère Hakim et a ma Chère Sœur Sarah.

A mes Chères Tantes, leurs Maris respectifs et plus particulièrement le Docteur ATROUCHE Omar pour sa présence et son aide, mes cousins et cousines.

A mon Oncle Mourad, sa Femme et la petite Nada.

A toute ma famille.

A mon cher Ami et Binôme Sofiane pour tous les moments de joies et de peines qu'on a passé ensemble, A sa Famille aussi.

A celle que j'aime.

A mes chers Amis : Halim, Mohamed, Fateh et Redouane.

A tous ceux qui ont supporté mes sauts d'humeur durant mes années universitaires et avec lesquels (lles) j'ai partagé mes souvenirs.

Je termine mes dédicaces en disant merci a mon promoteur BOUSTIA Narhimene.

A tous ceux qui ont cru en moi merci.

Lies

SOMMAIRE

Liste des Figures

INTRODUCTION GÉNÉRALE

Vu le statut important de la SONATRACH sur le plan national et international, l'entreprise donne une grande importance à son développement organisationnel et informationnel.

Pour cela, la direction de la SONATRACH à pour objectif d'assurer une bonne gestion de l'information et de la communication.

Du fait qu'il n'est pas toujours facile de retrouver l'emplacement exact d'une personne ou d'un matériel donné à l'intérieur de l'entreprise, d'avoir les informations élémentaires fiables et à temps réel, et encore moins en ce qui concerne les méthodes d'obtention de ces informations, l'obligation de développement d'une application de localisation se fait sentir.

Etant donné que l'informatique offre des solutions intéressantes permettant la maîtrise de l'information, de la communication et permet aux utilisateurs en général d'avoir une information fiable et à jour au moment opportun. Notre projet d'étude consiste en la réalisation d'un logiciel pour la gestion et la localisation géographique des bureaux du siège SONATRACH, ceci à travers la mise en place d'une base de données qui permettra la centralisation et la rapidité d'accès aux informations.

Pour ce fait, nous allons utiliser :

- La méthode OMT pour la conception.
- Le SGBD SQL Serveur 2000 pour l'implémentation de la base de données.
- CorelDraw, Microsoft Visuel Studio.Net, MapInfo, MapXtreme pour la réalisation du logiciel.

Introduction Générale

Notre document est reparti comme suit :

- Introduction générale.
- Présentation Générale de l'organisme d'accueil.
- L'étude de l'existant.
- Architecture multi-tiers.
- L'approche orientée objet.
- Analyse et conception.
- Implémentation et mise en œuvre : Présentation du logiciel.
- Conclusion et perspectives.

Chapitre I :

Présentation Générale

I.1. Présentation de l'organisme d'accueil SONATRACH:

I.1.1. Historique :

Consciente du rôle que devaient jouer les hydrocarbures dans la construction du pays, l'Algérie a décidé de prendre en main ce secteur juste après l'indépendance.

L'état algérien se dota d'un instrument permettant la mise en œuvre de sa politique énergétique en créant le 31 décembre 1963 par le décret n°63/491 la société nationale pour le transport et la commercialisation des hydrocarbures " **SONATRACH** ".

Les statuts de la société ayant été modifiés par le décret 66/292 du 22 septembre 1966, **SONATRACH** est devenue société nationale pour la recherche, la production, le transport et la commercialisation des hydrocarbures.

Le 24 février 1971, notre pays récupérait en toute légitimité sa souveraineté sur les richesses nationales en hydrocarbures et en confiant à **SONATRACH** la gestion et le développement.

Employant plus de 40000 personnes, **SONATRACH** se trouve aujourd'hui en position de jouer un rôle de premier plan et de consolider la position mondiale que lui confèrent :

- L'ampleur de la surface sédimentaire de l'Algérie.
- De services énergétiques.
- Des capacités technologiques et managements.

I.1.2. Les caractéristiques de SONATRACH :

Classée régulièrement parmi les 12 grandes entreprises pétrolières dans le monde, **SONATRACH** oriente ses actions notamment vers des missions stratégiques telle que :

- Le développement, l'exploitation et la gestion des réseaux de transport, de stockage et de chargement des hydrocarbures.
- La liquéfaction et le raffinage des hydrocarbures.
- Le développement de toutes formes d'activités conjointes en Algérie et hors Algérie avec des sociétés algériènnes ou étrangères.
- La prise et la détention de tout portefeuille d'action, les prises de participation et autres valeurs mobilières dans toutes sociétés existantes en Algérie ou à l'étranger.

- L'approvisionnement du pays en hydrocarbures à moyen et long terme.
- L'étude, la promotion et la valorisation de toute activité ayant un lien direct ou indirect avec l'industrie des hydrocarbures et de toute activité pouvant amplifie la tâche de la **SONATRACH**.
- La commercialisation des hydrocarbures.

I.1.3. Les objectifs de SONATRACH :

Les objectifs stratégiques de **SONATRACH** reposent sur :

- La maîtrise continue de ses métiers de base.
- Le renforcement des ses capacités technologiques et managériales.
- Le développement international et le partenariat.
- La diversification de son portefeuille d'activité.

I.1.4. L'organisation de SONATRACH :

L'entreprise **SONATRACH** s'articule autour des structures suivantes :

- La direction générale.
- Les structures opérationnelles.
- Les structures fonctionnelles.

I.1.4.1. La Direction Générale :

C'est la structure suprême de l'entreprise dirigée par le Président Directeur Général, nommé par décret présidentiel sur proposition du ministre chargé des hydrocarbures ; est investi des pouvoirs les plus étendus pour assurer l'administration, la gestion et la direction de **SONATRACH**.

Le PDG est responsable du fonctionnement général de la société, représente **SONATRACH** dans tout acte de la vie civile et exerce l'autorité hiérarchique sur le personnel.

I.1.4.2. Les Structures Opérationnelles :

Les structures opérationnelles sont organisées par **ACTIVITES**.

Chaque activité exerce ses métiers et développe son portefeuille d'affaires en national et en international.

- **Activité AMONT (AMT) :**

L'activité Amont, dotée de structures fonctionnelles communes, couvre notamment les domaines opérationnels suivants :

- Exploitation.
- Data contrôle.
- Opération.
- Affaires internationales.
- Recherche et développement des hydrocarbures et des nouvelles technologies.
- Production.
- Forage.
- Associations.

- **Activité Transport par Canalisations (TRC) :**

L'activité Transport par Canalisations dotée de structures fonctionnelles communes, couvre notamment les domaines opérationnels suivants :

- Exploitation des ouvrages de transport des hydrocarbures et des installations portuaires.
- Maintenance.
- Etudes et développement.

- **Activité Aval (AVL) :**

L'activité Aval, dotée de structures fonctionnelles communes, couvre notamment les domaines opérationnels suivants :

- Liquéfaction du gaz naturel.
- Séparation des GPL.
- Raffinage du pétrole.
- Pétrochimie.
- Etudes, développement et nouvelles technologies.

- **Activité Commercialisation (COM) :**

L'activité commercialisation, dotée de structures fonctionnelles communes, couvre notamment les domaines opérationnels suivants :

- Commercialisation extérieure du pétrole brut, des produits raffinés, du condensât, des GNL et du gaz naturel.
- Commercialisation sur le marché national des produits raffinés, du gaz naturel, des GPL, des produits pétrochimiques et des autres produits tels que les gaz industriels.
- Transport maritime des hydrocarbures.

I.1.4.3. Les Structures Fonctionnelles :

Les structures fonctionnelles ont pour rôle de :

- Assurer l'élaboration et la bonne application des politiques et stratégies du groupe.
- Planifier, fournir et cordonner la mise à disposition de l'expertise et l'appui aux différentes activités opérationnelles du groupe.
- S'affirmer en un centre d'excellence et d'expertise dans leurs domaines respectifs.
- Se constituer en centre d'information du groupe et contribuer au reporting général du groupe.

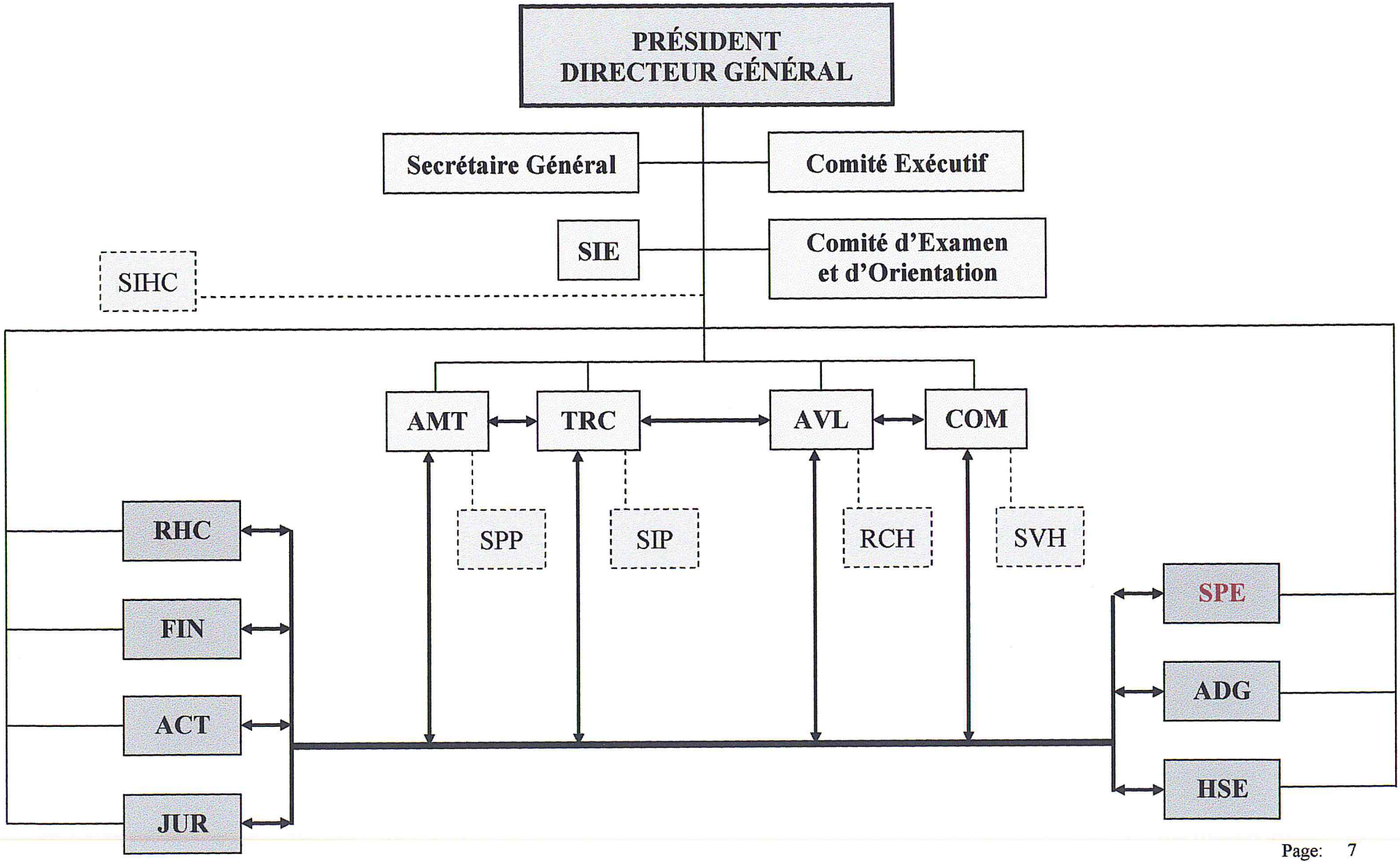
Les structures fonctionnelles sont organisées en quatre **Direction Coordination Groupe** qui sont :

- La Direction Coordination Groupe **R**essources **H**umaines et **C**ommunication « **RHC** »
- La Direction Coordination Groupe **S**tratégie, **P**lanification et **E**conomie « **SPE** »
- Coordination Groupe **F**inances « **FIN** »
- La Direction Coordination Groupe **A**ctivités Centrales « **ACT** ».

Et en trois **Directions Centrales** qui sont :

- La Direction Centrale **A**udit **G**roupe « **ADG** »
- La Direction Centrale **J**uridique « **JUR** »
- La Direction Centrale Santé, Sécurité et Environnement (**H**ealth, **S**afety & **E**nvironment) « **HSE** ».

SCHÉMA ORMANISATIONNEL ET FONCTIONNEL DE LA MACROSTRUCTURE SONATRACH



I.1.5. Les perspectives de SONATRACH :

Dans une économie mondiale en pleine mutation, l'engagement du processus de modernisation de SONATRACH s'impose inéluctablement pour assurer la pérennité de l'entreprise dans un environnement international plus que jamais concurrentiel.

La mise en œuvre de cette modernisation est d'autant plus impérieuse que l'entreprise a la charge de réaliser des objectifs déterminants pour l'avenir de la nation :

- Assurer la couverture à long terme des besoins nationaux en hydrocarbures : sa réalisation nécessite la constitution d'associations et la mise en œuvre d'accords au niveau de tous les gisements dans le pays ou à l'étranger.

- Contribuer à l'apport en devise pour le financement du développement économique de la nation : qui peut-être réalisé par l'accroissement des volumes commercialisés en devises, provenant des réserves nationales et à l'avenir d'autres réserves sur lesquelles **SONATRACH** peut prétendre aux droits de production ou bien par une meilleure valorisation des marchés extérieurs.

Prenant acte des évolutions en cours au sein du marché mondial des hydrocarbures, le projet de modernisation suppose un développement technologique, allié à un renforcement structurel et managérial.

I.2. Présentation de la structure d'accueil SPE :

I.2.1. La Direction Coordination Groupe Stratégie, Planification et Economie :

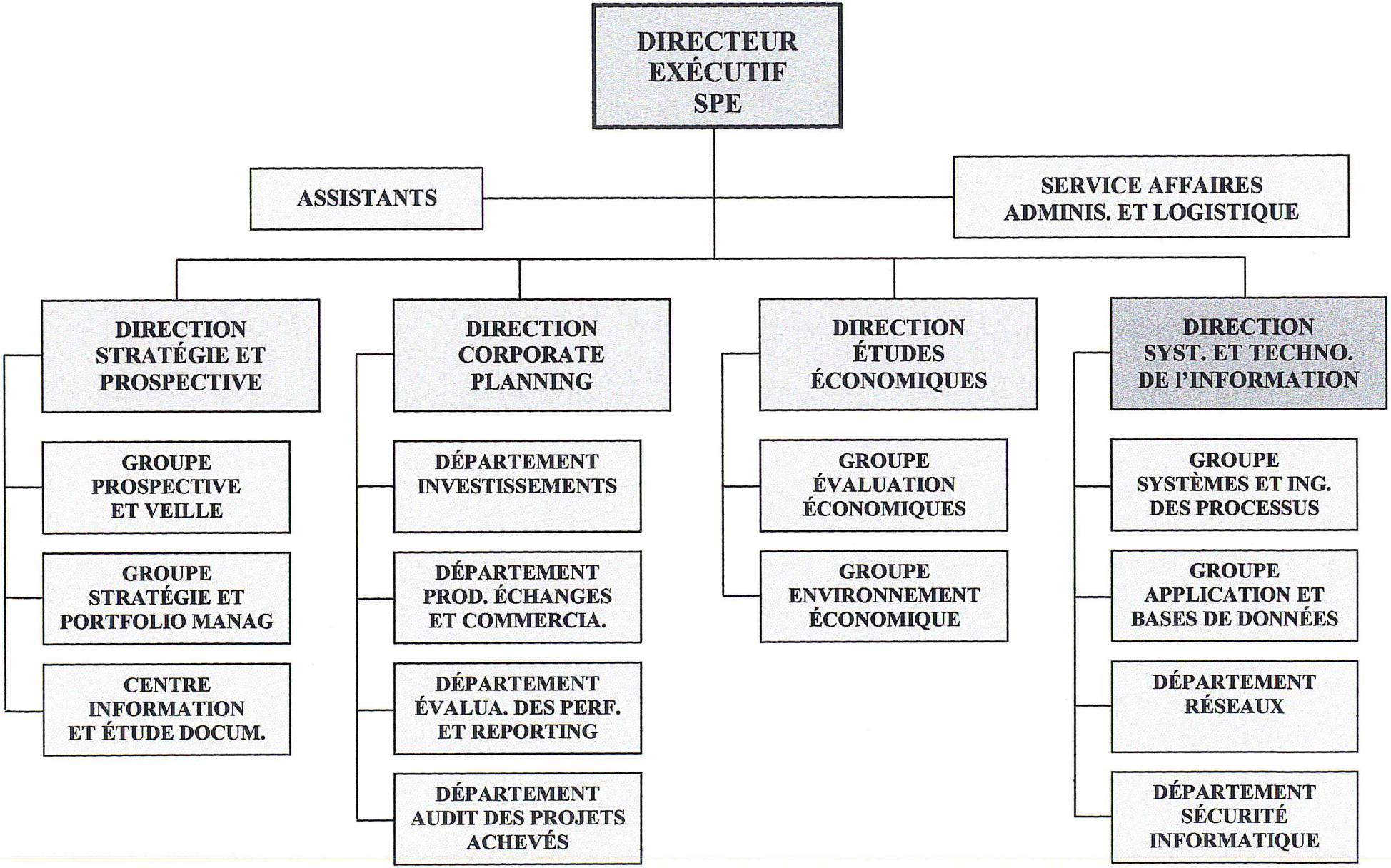
La direction Coordination Groupe Stratégie, Planification et Economie (**SPE**) est placée sous l'autorité d'un directeur exécutif chargé de l'élaboration des stratégies de développement à moyen et long terme et de l'évaluation de leur mise en œuvre.

I.2.2. Organisation de la SPE :

La Direction Coordination Groupe Stratégie, Planification et Economie (SPE) est organisée comme suit :

- **Direction Stratégie et Prospective,** composée de :
  - Un groupe spécialisé chargé de prospective et veille;
  - Un groupe spécialisé chargé de stratégie et portfolio management;
  - Un centre d'information et d'études documentaires.
- **Direction Corporation Planning,** composée de:
  - Un département investissement;
  - Un département production, échanges et commercialisation;
  - Un département évaluation des performances et reporting;
  - Un département audit projets achevés.
- **Direction Etudes Economiques,** composée de :
  - Un groupe spécialisé chargé des évaluations économiques;
  - Un groupe spécialisé chargé de l'environnement économique.
- **Direction Systèmes et Technologies de l'Information,** composée de :
  - Un groupe spécialisé chargé des systèmes et ingénierie des processus;
  - Un groupe spécialisé chargé des applications et bases de données;
  - Un département réseaux;
  - Un département sécurité informatique.
- **Service Affaires Administratives et Logistique**

ORGANIGRAMME DIRECTION COORDINATION GROUPE STRATÉGIE PLANIFICATION ET ÉCONOMIE



I.2.3 Missions essentielles de la SPE :

La Direction Coordination Groupe Stratégie, Planification et Economie a pour missions essentielles :

- L'animation et la coordination du processus de formulation, d'adoption et de suivi de la mise en œuvre de la stratégie du groupe.
- L'organisation, l'animation et la coordination du processus de planification de la société, en particulier le plan à moyen terme et le plan annuel, et pour les volets flux, échanges et investissements.
- La définition des objectifs, des politiques et des méthodes en matière de benchmarking.
- L'élaboration des politiques des normes et des standards de la société en matière de systèmes d'informations et de technologies de l'information.
- L'organisation des structures de la société.
- L'assistance aux activités dans le domaine de la stratégie, de l'analyse de portefeuille, d'études et de planification.
- La constitution de centre d'expertise dans chacun des domaines.
- La participation à toutes études stratégiques sur l'évolution du marché de l'énergie à moyen et long terme.
- Le développement, en relation avec les institutions universitaires et les organismes spécialisés, de nouvelles méthodes en terme d'évaluation économique, de management de risque et d'optimisation de portefeuille.
- L'organisation d'une conférence annuelle de ses cadres pour débattre des questions présentant un intérêt majeur en rapport avec les missions de la Direction Coordination Groupe Stratégie, Planification et Economie dont le thème et le contenu seront présentés lors de la conférence des cadres de **SONATRACH**.
- Le reporting à la direction générale sur l'organisation et les résultats de cette conférence.

I.3. Présentation du sujet

I.3.1. Problématique

Du fait qu'il n'est pas toujours facile d'avoir les informations élémentaires fiables et au moment voulu, et qu'il ne l'est encore moins en ce qui concerne les méthodes d'obtention de ces informations, l'obligation de développement d'applications propres aux besoins se fait sentir.

C'est dans se sens que la Direction Générale de SONATRACH se voit approprier une nouvelle application répondant à ses exigences et obligations, autrement dit : pouvoir localiser à tout instant un employé et un matériel qu'il soit informatique ou bureautique.

Nous pouvons recenser au niveau de la Direction Générale les insuffisances suivantes :

- Aucune application existante concernant la localisation géographique des employés et du matériel au niveau du siège.

- L'impossibilité de savoir en temps réel l'emplacement exact d'un matériel qu'il soit bureautique ou informatique au niveau des bureaux.

- La difficulté de localiser le bureau d'un employé donné.

I.3.2. Objectif de la nouvelle application

Afin d'assouvir les exigences de la Direction Générale de la SONATRACH, notre projet revient à concevoir et à réaliser une application qui répond aux besoins suivants :

- Mettre en place une base de données géographique et informationnel nécessaire à la localisation des employés et du matériel.

- Définir une architecture complète des blocs du siège, en précisant les zones et l'emplacement des bureaux.

Chapitre II :

Etude de l'Existant

II. Etude de l'existant

II.1. INTRODUCTION

Toute l'imagination et tout le savoir-faire de l'analyste seront sans emploi, s'ils ne peuvent s'exercer sur une base réaliste.

Pour que l'application soit mise à la disposition de ceux qui ont besoin et à qui elle est destinée, **l'étude de l'existant** est le point de passage obligé qui matérialise le premier contact avec un domaine ignoré, c'est la première étape dans notre étude, elle nous permet de découvrir -en détail- le domaine où la société souhaite améliorer son fonctionnement.

Cette étude a pour but de :

1/ Connaître en détail le domaine pour améliorer son fonctionnement.

2/ Examiner l'ensemble exhaustif des objectifs de ce domaine.

L'approche adoptée dans ce chapitre sera comme suit :

Entreprendre des enquêtes à base d'interviews auprès des responsables des services concernés par l'application, afin d'aboutir après analyse des données manipulées à une documentation complète du système actuel.

Enfin, on exposera les résultats du diagnostic du système existant sous forme de critiques et de suggestions qui permettront ainsi de juger de la nécessité du recours à l'informatique.

II.2. CODIFICATION

II.2.1. Définitions

Un code est une forme abrégée qui identifie un objet, il doit signifier l'information sans ambiguïté.

La relation qui fait la correspondance entre l'objet et son identifiant est appelée "codification", elle sert à passer de l'information sous sa forme naturelle à son symbole, tout en optimisant le système.

II.2.2. Règles

Les règles que doit vérifier une codification sont :

- Un code désigne un et un seul objet.
- Un code doit être le plus significatif possible.
- Avoir une codification facile et simple.
- La codification doit permettre la réduction de l'espace et le gain du temps.

II.2.3. Codification existante : On a remarqué la codification des objets suivants :

1- *Employé :*

Code alphanumérique sur six positions :

Exemple : 54872A

2- Structure :

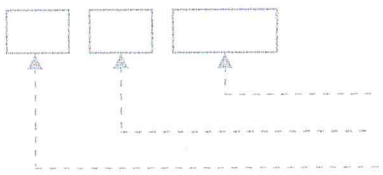
Code alphabétique sur trois positions :

Exemple : SPE : La Direction Coordination Groupe Stratégie, Planification et Economie.

3- *Bureau :*

Code alphanumérique sur quatre positions :

- *Signification de la codification :*



Numéro du bureau (02 numériques)
Numéro du niveau (01 numérique)
Le bloc (01 alphabétique)

Exemple : B804 : Le bureau numéro 04 du niveau 8 (Etage8) du bloc B.

4- *Matériel :*

L'objet " Matériel " a un code de taille non fixe.

Exemple : DF17H8WR903371K : Moniteur SAMSUNG 17".

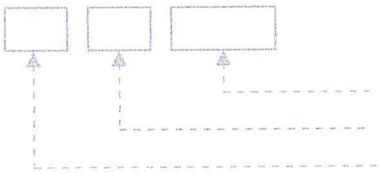
014136687 : Unité centrale, Pentium 4.

NB : Le matériel est codifié par le numéro de référence correspondant.

5- *Ligne téléphonique :*

Code numérique sur quatre positions :

- *Signification de la codification :*



Numéro chronologique (02 numériques)
Numéro du niveau (01 numérique)
Nature du propriétaire (01 alphabétique)

Nature du propriétaire	Désignation
5	Cadre
4	Secrétariat
3	Autres

Exemple :

5804 : Ligne affectée à un cadre ayant un bureau au 8ème étage avec un numéro chronologique de l'ordre 04.

II.2.4. Codification proposée

- *Employé:*

Nous allons garder la même codification en supprimant la position alphabétique du fait qu'elle n'a plus de signification.

Code numérique sur cinq positions:

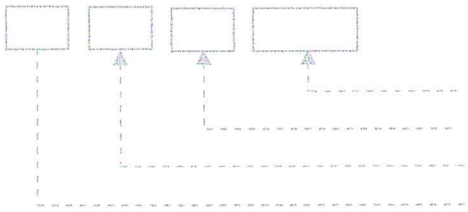
Exemple: 54872

- *Bureaux :*

Etant donné que chaque bloc est composé de zones, c'est pour cela qu'on introduit le numéro de la zone dans la codification du bureau, ainsi on aura la codification suivante :

Code alphanumérique sur cinq positions :

- *Signification de la codification :*



Numéro du bureau (02 numériques)
Numéro de la zone (01 numérique)
Le niveau (01 numérique)
Le bloc (01 alphabétique)

Exemple : B8604 : Le bureau numéro 04 de la zone 6 du niveau 8 (Etage8) du bloc B.

II.3. SOLUTION INFORMATIQUE

La solution informatique proposée consiste à exploiter les moyens existants au niveau de SONATRACH à savoir

- Le réseau INTRANET.
- Les micro-ordinateurs déjà connectés au réseau.

Pour la mise en place de cette solution, on a choisi de concevoir notre application en utilisant une architecture multi tiers. Pour cela, on a trouvé nécessaire de la présenter au chapitre suivant.

II.4. CONCLUSION

Cette étude nous a permis d'avoir un aperçu de la codification existante et proposer une solution informatique adéquate aux moyens existants au niveau de la SONATRACH.

Chapitre III :

Architectures Multi-tiers

III.1. Introduction

L'objectif premier d'un système d'information quel qu'il soit est de permettre à plusieurs utilisateurs d'accéder aux mêmes informations. Pour cela il faut donc regrouper les informations utilisées par l'entreprise. En terme technique, cela se traduit par la centralisation des données au sein d'une base de données. L'évolution des systèmes d'information s'est donc basée sur une meilleure subdivision entre les tâches à réaliser pour permettre l'exploitation de ces données par les utilisateurs finaux. Ceci permet de structurer plus efficacement les informations ce qui entraîne à la fois une meilleure organisation de l'entreprise et une meilleure efficacité technique. Cette subdivision a été facilitée par l'avènement des technologies orientées objets qui s'appliquent aussi bien au modèle client-serveur qu'au modèle Internet. Ces technologies permettent une séparation entre les différents composants du système. Il devient alors possible de réaliser de nouvelles architectures permettant la mise à disposition des informations sous différentes formes tout en diminuant les temps de développement. Ces technologies permettent également de faire collaborer une grande diversité de systèmes. On parle alors d'architecture distribuée. Il est ainsi possible de présenter des données en provenance d'un mainframe mélangées à des données en provenance d'un SGBDR, le tout étant affiché dans un browser sur la même page HTML.

La mise en oeuvre du modèle Client/Serveur, telle qu'elle a été faite jusqu'à présent par les grandes entreprises, n'a généralement pas été un véritable succès.

En effet, ce modèle très séduisant dans son concept et simple à utiliser dans un réseau local, est complexe à mettre en oeuvre à l'échelle de l'entreprise. Les effets " volume " (nombre et dispersion des postes de travail, importance du débit transactionnel, taille des bases de données, nombre d'utilisateurs simultanés accédant à la même base,...) font que le Client/Serveur " dit de 1ère génération " est totalement inadapté quand on cherche à l'étendre à l'ensemble de l'entreprise, et au delà à ses partenaires commerciaux.

De plus, la compétence et surtout l'expérience sont dans ce domaine encore rares. En conséquence, les grands projets Client/Serveur ont généralement une durée et des coûts sensiblement plus élevés que les estimations initiales (quand ils aboutissent). Fréquemment, ils ne s'intègrent pas (ou mal) dans le Système d'Information existant dans l'entreprise, ce qui à terme risque d'affaiblir la cohérence globale de celui-ci et surtout d'en augmenter son coût.

Le développement rapide et l'adoption par tous les acteurs majeurs du marché des technologies Internet/Intranet font émerger un nouveau modèle d'architecture Client/Serveur (dit " multi-tiers "), dont la caractéristique principale est de renforcer le rôle du réseau et des serveurs d'applications au détriment de celui des postes de travail (" Thin Clients " versus " Fat Clients ").

L'intérêt majeur de ce modèle est que, par construction, il combine les avantages des systèmes centralisés (cohérence globale, simplicité de développement, d'exploitation et de maintenance, maîtrise des coûts,...) et ceux du Client/Serveur (meilleure ergonomie du poste de travail, bonne intégration avec les outils bureautiques ou les autres applications, gains de productivité,...). Sa cible est la mise en place du Client/Serveur à l'échelle de l'entreprise, comme fondement technique de son Système d'Information. Il facilite aussi l'ouverture de celui-ci vers l'extérieur (clients, fournisseurs, partenaires, ...) via l'Internet.

Cette mise en oeuvre du Client/Serveur à grande échelle est grandement facilitée par l'utilisation des moniteurs TP qui isolent les développeurs d'applications de la complexité des mécanismes d'échanges distribués et transactionnels et qui donnent les moyens d'exploiter toute la souplesse de l'architecture " multi-tiers ".

III.2. Trois tâches importantes :

Tout système d'information nécessite la réalisation de trois groupes de fonctions: le stockage des données, la logique applicative et la présentation. Ces trois parties sont indépendantes les unes des autres: on peut ainsi vouloir modifier la présentation sans modifier la logique applicative. La conception de chaque partie doit également être indépendante, toutefois la conception de la couche la plus basse est utilisée dans la couche d'au dessus. Ainsi la conception de la logique applicative se base sur le modèle de données, alors que la conception de la présentation dépend de la logique applicative.

Stockage et accès aux données

Le système de stockage des données a pour but de conserver une quantité plus ou moins importantes de données de façon structurée. On peut utiliser pour cette partie des systèmes très variés qui peuvent être des systèmes de fichiers, des mainframes, des systèmes de bases

de données relationnelles, etc... Le point commun entre toutes ces systèmes est qu'il permettent le partage des données qu'ils contiennent via un réseau. La méthode d'accès à ces données dépendra du type d'organisation de ces données. Dans le cas d'une base de données relationnelle, l'accès peut se faire par des API qui dépendent du langage et de l'environnement. Ainsi en JAVA l'accès se fait via JDBC, alors qu'en C++ il se fera à l'aide d'ODBC. Quel que soit l'API, le langage SQL est utilisé.

Logique applicative

La logique applicative est la réalisation informatique du mode de fonctionnement de l'entreprise. Cette logique constitue les traitements nécessaires sur l'information afin de la rendre exploitable par chaque utilisateur. Les utilisateurs peuvent avoir des besoins très variés et évolutifs. Il devient alors nécessaire de permettre l'évolution du système sans pour autant devoir tout reconstruire. Cette partie utilise les données pour les présenter de façon exploitable par l'utilisateur. Il convient donc de bien identifier les besoins des utilisateurs afin de réaliser une logique applicative utile tout en structurant les données utilisées.

Présentation

La présentation est la partie la plus immédiatement visible pour l'utilisateur. Elle a donc une importance primordiale pour rendre attrayante l'utilisation de l'informatique. Son évolution a été très importante depuis les débuts de l'informatique. Depuis les terminaux en mode texte connectés à des mainframes jusqu'au HTML de nos jours en passant par les applications graphiques développées en client serveur, il y a eu beaucoup de chemin parcouru. Différents types d'interfaces demeurent intéressantes. En effet l'ergonomie d'un site Intranet HTML n'est pas forcément idéale pour tous les types d'applications. Il peut être intéressant de proposer plusieurs types d'interface pour une seule logique applicative. Par exemple une entreprise disposant d'un site de commerce électronique peut proposer un accès à la liste de ses produits sur Internet en HTML mais disposer d'une interface d'administration réalisée à l'aide d'une applet graphique.

III.3. Architecture client-serveur :

Les architectures client-serveur constituent une étape importante dans l'évolution des systèmes d'informations. Ce type d'architecture est constitué uniquement de deux parties: un **client** qui gère la présentation et la logique applicative, un **serveur** qui stocke les données de façon cohérente et gère éventuellement une partie de la logique applicative. L'interface entre ces deux parties est classiquement le langage SQL (dans sa version normalisée ou dans une version propriétaire). Dans ce type d'architecture on constate une certaine indépendance du client par rapport au serveur. La programmation du client peut s'effectuer sans se préoccuper de la mise en place de la base de données. En particulier, les questions concernant l'administration des données ne concerneront pas le développeur du client (dimensionnement des disques, répartition des données sur le système, optimisation des accès aux données, sauvegarde des données,...). Un des objectifs des architectures multi-tiers sera de généraliser ce type de fractionnement en éléments logiciels indépendants les uns des autres. La qualité de ce type d'architecture est basé sur deux technologies éprouvées: la notion de base de données relationnelle et le langage SQL. Toutefois on peut également constater deux inconvénients à cette technologie: elle possède une sécurité limitée et surtout son déploiement demeure long et laborieux.

Avantages: technologies éprouvées

Le modèle relationnel a été introduit par E. Codd en 1970. Il s'appuie sur des considérations théoriques formulées sous la forme de 12 règles formant le cahier des charges initial. Ce modèle est basé sur la notion de relation. Une relation est un ensemble de n-uplet (n est fixe) qui correspondent chacun à une propriété de l'objet à décrire. Ce modèle permet la gestion précise de contraintes d'intégrité qui garantissent la cohérence des données. Les systèmes de gestion de base de données relationnel (SGBDR) sont interfacés à l'aide d'un langage unique: le SQL (Structured Query Language). Ce langage permet d'effectuer l'ensemble des opérations nécessaires sur la base de données. Ce langage permet également la gestion de transaction. Une transaction est définie par quatre propriétés essentielles: Atomicité, Cohérence, Isolation et Durabilité (ACID). Ces propriétés garantissent l'intégrité des données dans un environnement multi-utilisateurs. L'**Atomicité** permet à la transaction d'avoir un comportement indivisible; soit toutes les modifications sur les données effectuées dans la

transaction sont effectives, soit aucune n'est réalisée. On comprend l'intérêt de ce concept dans l'exemple simple d'une transaction débitant un compte A et créditant un autre compte B : il est clair que la transaction n'a de sens que si les deux opérations sont menées à leur terme; L'atomicité de la transaction garantit que la base de donnée passera d'un état cohérent à un autre état cohérent. La **Cohérence** des données de la base est donc permanente. L'**Isolation** des transactions signifie que les modifications effectuées au cours d'une transaction ne sont visibles que par l'utilisateur qui effectue cette transaction. Au cours de la transaction, l'utilisateur pourra voir des modifications en cours qui rendent la base apparemment incohérente, mais ces modifications ne sont pas visibles par les autres et ne le seront qu'à la fin de la transaction si celle-ci est correcte (elle ne rend pas la base incohérente). La **Durabilité** garanti la stabilité de l'effet d'une transaction dans le temps, même en cas de problèmes graves tel que la perte d'un disque. L'ensemble de ces notions sont à la base des systèmes d'information actuels. Les technologies supportant ces notions (en particulier les SGBDR) forment l'aboutissement des recherches des 30 dernières années. Des systèmes comme Oracle offrent désormais des performances intéressantes y compris sur de grosses bases de données. Ces performances ont été obtenues sans faire de compromis sur les concepts théoriques sur lesquels ils se basent (modèle relationnel). Ces systèmes servent donc tout naturellement de base aux futurs systèmes d'information utilisant les nouvelles technologies Internet.

Inconvénients

L'architecture client-serveur possède toutefois des inconvénients. Ce sont ces inconvénients qui poussent les entreprises à utiliser d'autres technologies. Les deux inconvénients principaux sont la difficulté à gérer correctement les questions de sécurité et le coût du déploiement. La sécurité d'un système en architecture client-serveur est gérée au niveau du SGBDR. Celui-ci contrôle l'accès aux données en attribuant des autorisations d'accès aux différents utilisateurs du système. Le problème vient du fait que cette attribution de droit ne peut pas tenir compte des spécificités du logiciel réalisé. Pour pouvoir gérer les droits d'accès de cette façon il faut accorder à chaque utilisateur des droits sur un grand nombre de tables, ce qui devient vite laborieux. En pratique il est souvent plus rapide de ne créer qu'un utilisateur sur le SGBDR avec lequel tous les utilisateurs se connectent. Cette approche ne permet aucune gestion de la sécurité. Il suffit de connaître le login et le mot de passe d'accès à la base pour accéder à

l'ensemble des données sans aucune restriction. L'autre problème est souvent considéré comme beaucoup plus important par les entreprises car il est beaucoup plus visible. Il s'agit des durées et coûts de déploiement des logiciels. En effet un logiciel classique, développé en architecture client-serveur, nécessite une installation et une éventuelle configuration sur chaque poste utilisateur. Le déplacement d'un technicien coûte déjà très cher aux entreprises. Mais ce qui reste le plus laborieux est la nécessité de mettre à jour régulièrement le logiciel. Dans une architecture client-serveur, chaque mise à jour du logiciel nécessite un nouveau déploiement accompagné de nombreux coûts.

III.4. Architecture trois-tiers :

Le principe d'une architecture trois-tiers est relativement simple: il consiste à séparer la réalisation des trois parties vues précédemment (stockage des données, logique applicative, présentation). Nous avons déjà pu entrevoir la possibilité de séparer la conception de ces trois subdivisions, ici il s'agit de séparer leur implantation. Tout comme dans le client-serveur cette séparation signifie qu'il est possible de déployer chaque partie sur un serveur indépendant, toutefois cela n'est pas obligatoire. La mise en place de ce type d'architecture permet dans tous les cas une plus grande évolutivité du système. Il est ainsi possible de commencer par déployer les deux serveurs sur la même machine, puis de déplacer le serveur applicatif sur une autre machine lorsque la charge devient excessive. Les éléments permettant la réalisation classique d'un système en architecture trois tiers sont les suivants:

- système de base de donnée relationnel (SGBDR) pour le stockage des données
- serveur applicatif pour la logique applicative
- navigateur web pour la présentation

Il est important de remarquer que l'essentiel du travail de développement sera implanté au niveau du serveur applicatif. Le SGBDR nécessitera un travail d'administration surtout dans le cas d'une quantité de données importante. Le travail de conception de la base de donnée sera la pierre angulaire du système. En effet l'ensemble du développement s'appuiera sur cette conception. Le navigateur web nécessitera la programmation de code spécifique permettant de gérer l'affichage par ce navigateur. Ce code sera placé sur le serveur applicatif pour permettre une mise à jour sans nécessiter de nouveaux déploiements.

Avantages de cette architecture

Cette architecture se développe actuellement au sein des entreprises grâce aux nombreux avantages qu'elle présente. Malgré la différence évidente entre une architecture trois tiers et un système client-serveur (l'apparition d'un serveur pour la logique applicative), le système reste basé sur les technologies éprouvées détaillées précédemment (aspect relationnel et transaction). La logique applicative est déplacée au niveau du serveur d'application mais reste programmée à l'aide des mêmes technologies liées aux bases de données relationnelles. En particulier l'utilisation du langage SQL reste jusqu'à présent la solution la plus intéressante au niveau de la qualité logicielle. Elle présente à la fois une grande fiabilité, une bonne disponibilité, une excellente évolutivité,... Toutefois il faut prendre en compte deux facteurs importants: d'une part le choix du SGBDR (ils n'ont pas tous les même qualités), d'autre part la qualité des programmes utilisant la base de données (aussi bien au niveau de la conception que de la programmation). L'avantage principal d'une architecture multi-tiers est la facilité de déploiement. L'application en elle même n'est déployée que sur la partie serveur (serveur applicatif et serveur de base de données). Le client ne nécessite qu'une installation et une configuration minime. En effet il suffit d'installer un navigateur web compatible avec l'application pour que le client puisse accéder à l'application, ce navigateur étant par ailleurs souvent installé par défaut sur toutes les machines. Cette facilité de déploiement aura pour conséquence non seulement de réduire le coût de déploiement mais aussi de permettre une évolution régulière du système. Cette évolution ne nécessitera que la mise à jour de l'application sur le serveur applicatif. Ceci est très important car cette évolutivité est un des problèmes majeurs de l'informatique. Le troisième avantage est l'amélioration de la sécurité. Dans un système client-serveur tous les clients accédaient à la base de données ce qui la rendait vulnérable. Avec une architecture multi-tiers l'accès à la base n'est effectué que par le serveur applicatif. Ce serveur est le seul à connaître la façon de se connecter à cette base. Il ne partage aucune des informations permettant l'accès aux données, en particulier le login et le password de la base. Il est alors possible de gérer la sécurité au niveau de ce serveur applicatif, par exemple en maintenant la liste des utilisateurs avec leurs mots de passe ainsi que leurs droits d'accès aux fonctions du système. On peut même améliorer encore la sécurité par la mise en place d'une architecture réseau interdisant totalement l'accès au serveur de base de données pour les utilisateurs finaux. La mise en place de firewall correctement configuré permettra ceci. Dans le cas de la mise en place d'un workflow au sein de l'entreprise Kallisto,

l'avantage le plus intéressant est sans aucun doute l'évolutivité du système. En effet les besoins actuels sont relativement vastes, toutefois la réalisation totale semble relativement longue et difficile à réaliser d'un seul coup. De plus des problèmes risquent de nécessiter une mise à jour répétée du système.

III.5. Architecture multi-tiers

Nous avons vu ce qu'était une architecture trois-tiers, une architecture multi-tiers va plus loin dans le découpage de l'application sur différents serveurs. Une architecture multi-tiers est également appelé architecture distribuée du fait de la distribution des traitements et des données sur différents serveurs. Le découpage de base du système reste toujours le même: une partie gestion de données, une partie gestion de la logique applicative et bien entendu le client assurant la présentation. Toutefois les deux parties développées coté serveur vont pouvoir être déployées chacune sur plusieurs serveurs. L'objectif général de ce type d'architecture est de permettre l'évolutivité du système sous plusieurs aspects: la quantité de données stockée, la disponibilité du serveur, le nombre d'utilisateurs,... Il faut également bien comprendre qu'un système conçu en architecture multi-tiers n'est pas forcément déployé sur plusieurs machines dès le départ. Toutefois son mode de programmation doit permettre de modifier le déploiement du système en cours d'exploitation par un administrateur. Le développeur doit donc rendre le système indépendant du serveur sur lequel il s'exécute. Il existe deux types de répartition possibles dans une architecture distribuée. Il est possible de répartir les données et de répartir la logique applicative. Chacune de ces deux répartitions permet de résoudre des problèmes de natures différentes. Elles peuvent donc être mises en place soit séparément soit en parallèle sur le même système.

Répartition des données sur différents serveurs

Il s'agit d'utiliser plusieurs sources de données dans une même application. Chaque source possède sa propre structure de données. Chaque serveur de données peut être géré de façon très différente. Toutefois une interface de programmation commune à toutes les sources doit pouvoir exister. Il peut exister des relations entre les données des différents serveurs, il sera alors nécessaire de gérer des transactions distribuées entre des différents serveurs de données. Cette répartition de données correspond à la notion de base de données distribuée. Les bases de données distribuées permettent de résoudre deux types de problèmes. La première est la performance du système: la répartition des données permet d'augmenter la disponibilité des données. Toutefois il est nécessaire de bien penser cette répartition afin de ne pas démultiplier le nombre de transactions distribuées nécessaires. Ceci aurait pour effet de diminuer la performance plus que de l'augmenter. Le deuxième type de problèmes est la réutilisation des systèmes existants. En effet de nombreux systèmes informatiques stockent actuellement une grande quantité de données. Toutefois ces systèmes ne sont pas toujours bien coordonnés entre eux. Ceci risque d'entraîner des redondances dans les données et des incohérences entre plusieurs sources de données. L'objectif est donc de réutiliser ces systèmes dans une même application afin de restructurer le système d'information sans pour autant repartir de zéro (ce qui est non seulement un non sens mais est aussi impossible). Les systèmes réutilisés sont bien souvent hétéroclites (mainframe, SGBDR,...) et nécessitent d'être réunis en utilisant diverses technologies (moniteurs transactionnels, serveurs d'applications,...).

Répartition de la logique applicative

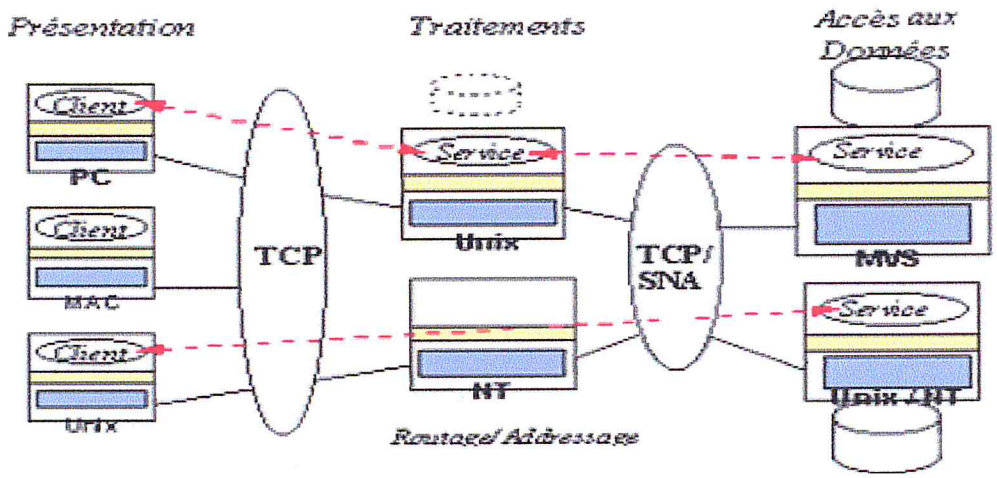
La répartition de la logique applicative permet de distribuer les traitements sur différentes machines. Cette distribution est facilitée par l'utilisation de la programmation orientée objet et plus particulièrement de ce qu'on appelle les composants. Un composant possède entre autre la caractéristique d'être accessible à travers le réseau. Un composant peut ainsi être instantié puis utilisé au travers du réseau. Il est également possible de trouver un serveur permettant l'utilisation d'un composant, ce qui permet une forte évolutivité ainsi qu'une résistance aux pannes importantes (le service sera toujours disponible sur un serveur même si une machine tombe en panne). La réalisation de ce type de répartition nécessite l'utilisation de technologies spécifiques. D'une part il faut permettre la communication entre les différents éléments de l'application (RMI), d'autre part le déploiement et l'évolution du système doivent pouvoir être

assurés (JNDI + fichiers XML). Certaines technologies proposent une architecture cohérente pour la mise en place des différentes technologies intervenant sur un système distribué (EJB). Les buts recherchés lors de la mise en place de ce type d'architecture sont la performance, l'évolutivité, la maintenabilité...

***Modèle Technique Général des architectures* " multi-tiers "**

Les applications constituant le Système d'Information de l'entreprise comportent généralement trois grands types de composants :

les fonctions de présentation (ou Interface Homme/Machine)
les traitements
l'accès aux données qui, dans l'architecture " multi-tiers " sont répartis/distribués sur plusieurs plates-formes suivant trois niveaux tels que représentés par le schéma ci-dessous.



Ce modèle distingue trois types de plates-formes :

les postes de travail (NC, PC, MAC, stations UNIX) qui assurent principalement les fonctions d'Interface Homme/Machine ainsi que les contrôles locaux ou les petits traitements associés à celles-ci.
les serveurs locaux (ou départementaux) qui assurent les communications entre les postes de travail et les serveurs centraux, et effectuent une partie des traitements, avec éventuellement accès à des données locales.
les serveurs centraux qui assurent qui assurent principalement l'accès aux données de l'entreprise et effectuent les traitements correspondants, selon les règles de gestion applicables à celles-ci.

Les architectures d'applications correspondantes à ce modèle technique général distinguent:

les modules de service ou plus simplement " les Services " qui sont de véritables " composants réutilisables " assurant l'encapsulation des données et des traitements associés sous forme " d'Objets Métier " et qui s'exécutent sur les serveurs centraux et éventuellement sur les serveurs locaux/départementaux
les programmes Client ou plus simplement " les Clients " qui gèrent le dialogue Homme/Machine sur les postes de travail et qui appellent les composants d'applications correspondants aux requêtes de service des utilisateurs.

Dans ce modèle, les serveurs locaux/départementaux peuvent être :

soit simplement des serveurs techniques assurant la mise en relation des Clients avec les Services, en fonction des requêtes émises par les utilisateurs,
soit des serveurs mixtes (techniques et applicatifs) supportant également une partie des traitements et éventuellement l'accès à des données locales.

Par exemple, lorsque le traitement d'une requête de l'utilisateur requiert l'appel à plusieurs services s'exécutant sur différents serveurs centraux, il est intéressant de développer un service plus global qui réalise, sur le serveur local/départemental auquel est connecté le poste client, " l'intégration " des services élémentaires nécessaires (on parle quelque fois de " services d'intégration " pour qualifier les composants s'exécutant dans les serveurs intermédiaires).

Inversement, quant une requête utilisateur se traduit simplement par l'appel d'un ou deux services, le client peut invoquer directement les composants applicatifs (les " Objets Métiers ") s'exécutant sur les serveurs centraux mais en utilisant néanmoins les couches de communication du serveur intermédiaire auquel est attaché le poste de travail.

Rôle du Middleware

La mise en oeuvre d'une architecture " multi-tiers " nécessite sur l'ensemble des plates-formes des logiciels génériques, appelés globalement " Middleware ", qui fournissent les mécanismes techniques supportant les échanges d'information entre Clients et Services.

Le " Middleware " se définit comme l'ensemble des logiciels mis en oeuvre " en dessous des applications, au dessus des O.S. et entre les plates-formes " pour assurer les échanges entre les composants Présentation, Traitement et Accès aux données dans les architectures Client/Serveur.

Les principales fonctions attendues du " Middleware " sont :

la localisation des Services (permettant l'adressage symbolique de ceux-ci, généralement sur la base d'un annuaire qui peut être centralisé ou distribué)
les échanges de messages entre Clients et Services, en mode synchrone (requête/réponse) ou asynchrone (via des files d'attente).
les mécanismes de formatage et conversion de données (requis du fait de l'hétérogénéité des plates-formes).
les fonctions de sécurité (authentification, contrôle d'accès, chiffrement,...)
l'administration (configuration et exploitation) de l'ensemble des composants (système et application) des différentes plates-formes, généralement à partir d'un point central unique.

Le choix du " Middleware " (et surtout sa mise en place) est dans les grandes entreprises qui ont en général un environnement informatique hétérogène, un problème complexe dont la solution passe le plus souvent par l'intégration de plusieurs produits d'origines différentes. Les Moniteurs TP, qui offrent en standard ce type de fonctionnalité, sont généralement un bonne solution à ce problème.

Avantages de l'architecture " multi-tiers "

Par construction, l'architecture " multi-tiers " telle que définie précédemment et basée en particulier sur les technologies Inter/Intranet, résout certaines difficultés intrinsèques au Client/Serveur " dit de première génération " :

La gestion des données reste centralisée, ce qui d'une part simplifie l'exploitation et d'autre part garantit la cohérence globale du Système d'Information.
La réalisation de composants applicatifs réutilisables (les services) sous forme d'objets métiers permet de simplifier la maintenance, et de diminuer les coûts de développement (on pourra construire de nouveaux services par assemblage de services existants).
l'utilisation des systèmes (MVS/IMS) et des bases (DL1/DB2) existants comme serveurs de données d'entreprise (" cohabitants " avec des serveurs de données UNIX) permet une réutilisation importante du " capital informatique " de l'entreprise et de la compétence de ses développeurs.
le concept de " Thin-Clients " réduits aux seules fonctions de présentation/contrôle et appelant systématiquement des " Services/Objets Métiers " facilite la maintenance des postes de travail et favorise la réutilisabilité des composants applicatifs, en apportant l'ergonomie des IHM graphiques et l'intégration avec les outils bureautiques.
l'usage d'un Browser comme client universel, adapté aux besoins des applications par des pages HTML dynamiques et/ou des Applets Java, assure l'indépendance vis à vis du type de poste de travail, simplifie et accélère le développement des Clients et surtout solutionne le difficile problème de mise à niveau de ces postes.
le rôle joué par les systèmes intermédiaires UNIX ou NT comme Serveur de Communication et éventuellement comme Serveur d'Intégration simplifie sensiblement le " Middleware " tout en optimisant l'usage des ressources réseau et système.
des postes Clients configurables dynamiquement et capables de fonctionner avec un bonne ergonomie et de bons temps de réponse sur Internet permettent d'ouvrir le système d'information de l'entreprise aux partenaires commerciaux

Chapitre IV :

Méthodes de Conceptions

IV.1. ETUDE DES METHODES DE CONCEPTION

1.1. Introduction :

Au fur et à mesure que les systèmes deviennent complexes, la nécessité de rationaliser leur conception et leur développement s'impose et vu que le coût de maintenance des systèmes est parfois plus élevé que leur développement. Donc, il est nécessaire de disposer de méthodes et des techniques de conception afin de diminuer ces coûts et limiter ces risques. Ces méthodes apportent principalement une démarche à un plan d'action, des méthodes de synthèse et des outils de contrôle de qualité.

1.2. Le rôle des méthodes :

« La méthode définit un ordre logique dans lequel les tâches doivent être réalisées pour atteindre un but défini. Elle spécifie l'inventaire, la nature et le contenu des tâches ainsi que l'ordre d'exécution et les résultats à l'issue de chaque tâche » **[RUM 91]**

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 visée. Ce fil conducteur permet de :

- Savoir quoi faire, et dans quel ordre.
- Savoir où on est réellement, ce qui reste à faire et où on va.

1.3. Les différentes classes de méthode de conception :

Dans l'évolution chronologique des méthodes de spécifications et de conception des systèmes d'information, trois générations de classes ont apparu :

Les approches cartésiennes (Première génération) :

La première génération de méthodes 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 soit simple à réaliser. On peut citer comme exemple : SADT, Jackson, …..

Les points forts de cette approche sont :

- La simplicité : elle traduit le bon sens et la démarche naturelle pour aborder un problème ;
- La capturation plus facile des besoins des utilisateurs.

Les points faibles :

- Une concentration de l'effort d'analyse sur les fonctions, c'est-à-dire une négligence de la cohérence des données en une 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.

Les approches systémiques (Deuxième génération) :

Dans cette deuxième génération, le système d'information est perçu comme un objet complexe actif dont il faut décrire la structure (où l'architecture) et les objectifs fonctionnels (les fonctions). Le système est abordé de point de vue modélisation des données et de point de vue modélisation des traitements. On peut citer comme exemple : Merise, Axial, …

Les points fort de cette génération :

- Une plus grande cohérence des données.
- Le respect des niveaux de conception introduits par rapport à la norme internationale (niveau externe, niveau conceptuel et le niveau interne).
- La capacité de décrire des systèmes complexes.

Comme points faibles de ces méthodes, on peut citer :

- 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 de traitements mélange la connaissance et le contrôle.
- Un cycle de développement incomplet pour les traitements (ce qui fait dire de ces méthodes que ce sont plus des méthodes de conception que des méthodes de développement).

Les approches orientées objet (Troisième génération) :

Cette approche est une évolution de la deuxième génération des méthodes 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 système d'information comme un ensemble d'opérations attachées aux objets composants ce système. Ceci donne une meilleure compréhension de la sémantique des objets et permet une meilleure modularité et réutilisabilité des composants. On peut citer comme exemple : OOD, HOOD, OMT, ...

Les points forts de cette approche :

- Une grande capacité à modéliser des objets complexes.
- La réduction des distorsions entre le monde réel et le système informatique.
- Un moyen d'exprimer d'une façon intégrée la dynamique des objets.
- Un moyen d'encapsuler les objets pour cacher leur implémentation et ne laisser voir que les services rendus.

Comme points faibles, on peut citer :

- C'est la philosophie des « tout objet ». Dans les organisations, l'aspect fonctionnel est très important, donc 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 concept à réaliser.

1.4. Les méthodes Orientées Objets :

S'il existe maintenant un certain nombre de méthodes de conception de logiciels par objets, il n'en existe par réellement une qui soit spécifiquement adaptée à la conception de bases de données objets. Nous allons donc passer en revue un certain nombre de méthodes de conception objets pour tenter d'en extraire les idées les plus intéressantes.

On pourra néanmoins trouver dans **[Arnold et Al 91]** une évaluation de plusieurs de ces méthodes et dans **[Wilke 93]** un résumé, une comparaison et des exemples des méthodes : Booch, Wirfs-Brock, Hood, Coad and Yourdon, Winter Partners, Shlaer and Mellor, Jacobson, Wasserman et Al, Runbaugh, Reenskang et Al, and Colbert.

Nous présenterons tout d'abord la méthode AOO qui est une méthode d'analyse. Sa démarche est très classique, mais elle est très clairement exprimée. Puis nous présenterons la méthode OOD qui est elle une méthode visant plus spécifiquement la phase de conception. Ces deux méthodes, bien que n'étant pas basées sur les mêmes notations, ont en fait un formalisme très proche l'une de l'autre et se complètent très bien. On verra par la suite la méthode HOOD et la méthode OMT.

1.4.1. La méthode d'Analyse Orientée Objets (AOO) :

L'analyse orientée objets est une méthode issue des travaux de P. Coad et E. Yourdon en 1990, et s'appuie sur trois grands principes de base permettant d'appréhender et d'organiser un modèle du monde réel :

- La différenciation et l'expérience en objets élémentaires et en leurs attributs.
- La distinction entre objets complets et leurs composants internes.
- La formation de différentes classes d'objets et la distinction entre ces classes.

La démarche de l'AOO :

La démarche de la méthode AOO, présentée dans **[RUM 91]**, est définie par cinq grandes étapes :

- Trouver les « classe et objet » : ce terme désigne une classe et les objets qu'elle contient.
- Identification des structures : cette étape décrit la structure de généralisation-spécialisation (appelée aussi structure de classification) et la structure de composé-composants.
- Identification des sujets : les sujets sont utiles pour organiser le travail en « lots » sur les gros projets.
- Définition des attributs : cette étape permet d'identifier les attributs, les placer dans des classes et objets qu'ils décrivent le mieux et identifier les connexions d'instances pour modéliser l'association.
- Définition des services : les services détaillent l'abstraction de la réalité à modéliser, en précisant le comportement de chaque objet d'une classe. Elle permet d'identifier les états de l'objet, les services nécessaires et les connexions de messages.

Nous allons maintenant détailler ces cinq étapes :

Identification des « classes et objets » : cette première phase a pour objet d'identifier les objets dans le contexte du problème à modéliser. L'identification des objets n'étant pas toujours un processus intuitif, cette méthode propose un certain nombre de conseils pour nous aider dans cette démarche.

- On doit rechercher les objets dans l'espace du problème à analyser. On étudiera donc tout d'abord le domaine général du problème. Pour cela les auteurs conseillent de demander aux utilisateurs un résumé sur le sujet. Ils conseillent également de consulter une encyclopédie ou des ouvrages de vulgarisations pour pouvoir apprendre la terminologie et principaux concepts du domaine étudié. Il faudra ensuite rechercher les objets dans les documents de spécification (textes, diagrammes) donnés par le client.
- Pour trouver les objets potentiels, on peut essayer de rechercher entre autres : des structures d'autres systèmes ou mécanismes, des événements enregistrés, des rôles joués, des localisations et des organisations.
- Ayant déterminé un certain nombre d'objets candidats pour la modélisation, quels sont les objets que l'on doit réellement prendre en compte et à partir de quels critères doit-on les choisir ?

  Si l'objet considéré possède plusieurs attributs potentiels qu'il est important de connaître, ou s'il est nécessaire de définir un ou plusieurs services à son niveau, alors cet objet doit être pris en compte dans la modélisation.

  Par contre, si l'objet ne semble avoir qu'un seul attribut, et ne pas nécessiter de services particuliers alors, il n'est en général pas nécessaire de prendre en compte cet objet en tant que tel, mais il doit plutôt être pris en compte en tant que simple attribut d'autres objets.
- Pour nommer les objets, on utilisera un nom, ou un couple nom+objectif. Le nom d'un objet doit décrire une occurrence unique d'un objet, plutôt qu'un ensemble d'objets. Pour faciliter le dialogue avec les utilisateurs, on utilisera le plus possible le vocabulaire du domaine modélisé.
- La notation utilisée regroupe dans une « boite » l'ensemble des informations sur l'objet.

Identifier les structures : les structures sont utilisées pour faire face à la complexité du domaine étudié. Elles aident à l'organiser pour pouvoir mieux l'appréhender. On distingue deux grands types de structures : les structures de classification et les structures de composition.

1- Structure de classification :

L'utilisation de structures de classification est un des principaux moyens utilisés pour organiser la connaissance sur un domaine d'étude. Elle permet le regroupement d'entités en classes. Ces classes étant elle-mêmes liées par des liens de type spécialisation et généralisation. Il est ainsi possible de regrouper ensemble des objets ayant le même type d'attributs et le même type de comportement pour chaque objet par rapport aux autres objets en tant que spécialisation ou généralisation de ces objets.

- On cherchera tout d'abord s'il existe des spécialisations de cet objet qui ont un sens pour l'étude. On cherchera donc des sous-types de l'objet, qui seront plus spécifiques que celui-ci par l'existence d'attributs ou de comportement supplémentaires.
- On tentera ensuite de considérer les objets candidats sous le point de vue inverse, pour chercher s'il n'existe pas un ou plusieurs objets plus généraux permettant de regrouper plusieurs objets ensemble car possédant des caractéristiques communes.

Les regroupements effectués devront avoir un sens dans le cadre de l'étude et ne pas seulement être dus à de simples similitudes entre objets.

2- Structure de composition :

L'utilisation de composition permet de représenter les objets sous la forme d'assemblages de type ensemble / sous-ensemble. Il est ainsi possible de percevoir un objet, soit comme un tout, soit comme étant constitué lui même d'autres sous-objets qui peuvent eux mêmes être composés d'autres éléments. Ce type de structure permet donc de modéliser naturellement l'agrégation de composants dans un assemblage. Pour chaque objet candidat de la modélisation, on considérera tour à tour cet objet comme étant lui-même composé d'autres objets ou, au contraire, comme étant un constituant d'un ensemble plus grand.

- Pour chaque objet, on cherchera tout d'abord à le considérer en tant qu'assemblage de composants plus élémentaires. Si de tels composants existent, ont-ils une importance et un sens au niveau de l'étude ? Dans ce cas il faudra les caractériser plus complètement.
- Inversement, on cherchera également si un objet ne peut pas être considéré comme étant un composant d'un ensemble ou d'une organisation plus importante. Si une telle structure existe et fait partie du sujet de la modélisation, il faudra la prendre en compte.

Définir les sujets : si l'espace du problème étudié est important, et donc comporte un grand nombre d'objets, on va rapidement se heurter à des difficultés pour percevoir de façon globale l'ensemble du modèle. La définition de sujets est donc un moyen de structurer le modèle en des sous- modèles organisés hiérarchiquement. Cette décomposition en sujets permettra d'avoir une vue d'ensemble du problème. Elle permet également une bonne communication entre les différents acteurs de la modélisation en limitant le nombre d'objets à considérer en même temps.

Pour un projet de taille moyenne (quelques dizaines d'objets), on pourra déterminer les sujets après coup, après avoir identifié les structures. Pour des projets de plus grande importance, on cherchera rapidement à identifier les sujets, pour permettre un découpage de l'analyse.
Définir les attributs et les liens : les attributs permettent de décrire plus complètement un objet. En définissant les attributs d'un objet, on affine l'analyse, en détaillant quelles sont les propriétés de l'objet que l'on doit prendre en compte et qui ont un sens dans le cadre du problème étudié. A ce stade de l'analyse, il est important de ne choisir que les attributs "pertinents". Par exemple, un attribut couleur pour un outillage de moulage n'a pas à être pris en compte, sauf si cette couleur possède une signification particulière dans l'atelier.
De même, pour chaque attribut, il faut se demander si cet attribut est réellement élémentaire, ou il ne référence pas au contraire un objet qu'il faut modéliser en tant que tel. Il pourra éventuellement être bon de repousser ce choix à la phase de conception proprement dite.

Si l'analyse a fait apparaître des structures de classification, on cherchera à mettre en facteur les attributs communs à plusieurs objets au niveau de leur généralisation commune.

C'est également à cette étape que l'on va définir les relations ou liens existants entre les différents objets. Ces liens servent à représenter les relations autres que celles de classification

ou de composition qui, nous l'avons vu, ont été identifiées dans l'étape précédente. On établira donc tout d'abord les connexions entre les différents objets, puis, pour chaque lien, on déterminera la multiplicité et la participation de cette relation.

La multiplicité d'un lien exprime le nombre de successeurs de ce lien : soit il n'y a qu'un seul successeur et le lien est de type 1 :1 ; soit le lien représente en fait l'association d'un objet avec un ensemble d'autres objets et alors le lien est de type 1 : M.

Définir les Services : la dernière étape de l'analyse orientée objet consiste à définir les différents services attachés aux objets. Les services permettent de modéliser le comportement de chaque objet. Ils permettent également de spécifier les interactions de l'utilisateur avec le système et celles entre les objets du système.

Identification des services : une première stratégie pour identifier les différents services consiste à considérer chaque objet suivant l'un des trois points de vue suivants :

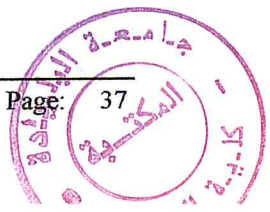
- Actions directes sur l'objet (création, modélisation, sélection, destruction, ….).
- Calculs effectués par l'objet.
- Attente d'événements (monitoring).

Le premier type (actions directes sur les objets) est tellement commun, qu'il est considéré comme implicite pour tous les objets. On ne le spécifiera que s'il présente des particularités au niveau d'un objet donné.

Une seconde stratégie pour identifier d'autres services est de considérer l'histoire de chaque objet. Au cœur de l'existence de l'objet (entre sa création et sa destruction), existe-t-il des actions autres que de le modifier ou de le sélectionner ? De telles actions sont alors à prendre en compte.

Une dernière stratégie est de considérer l'ensemble du système et de des états, et de voir à quels événements extérieurs il doit répondre. On établira donc un diagramme ou une table / événement / transition. On identifiera alors quels sont les objets du système concerné par ces événements.

Identification des messages : une fois les services identifiés, on détaillera les échanges entre les différents objets sous la forme de connections entre ces objets. De même on fera apparaître les commandes provenant de l'utilisateur vers les objectifs concernés.



Spécification des services : on spécifiera chaque service, en prenant bien soin de considérer ce service uniquement du point de vue du comportement externe de l'objet. Que demande-t-on à l'objet, et de quoi est-il responsable ?
La description du service se fera sous forme de texte décrivant le service.

1.4.2. La méthode OOD (Objet Oriented Design) :

La méthode OOD (appelée aussi méthode « Booch ») a été présentée initialement par G. Booch, puis a été complétée et améliorée. Elle constitue une méthode de développement de systèmes complexes.

Un concepteur de programmes informatiques doit considérer les questions fondamentales suivantes en conception Orientée Objet :

1) Quelles classes existent et comment sont-elles reliées ?
2) Quels mécanismes sont utilisés pour contrôler la collaboration entre objets ?
3) Où doit-on déclarer chaque classe ou chaque objet ?
4) A quel processeur, un processus doit-il être alloué, et comment les processus multiples d'un processeur donné doivent-ils être planifiés ?

G. Booch a établi dans plusieurs de ces articles et ces ouvrages sur les principales étapes conduisant à la conception orientée-objet. Ces étapes ont été reprises dans leurs grandes lignes par toutes les méthodologies parues depuis.

Identifier les objets et les classes : cette première étape vise à identifier les objets du monde réel que l'on voudra réaliser. Pour cela, on doit identifier les propriétés caractéristiques de l'objet. Cette étape est bien entendu celle qui demande le plus de talent et d'expérience personnelle. Un moyen relativement informel pour identifier les objets consiste à faire une description littéraire (en français) du problème. On pourra déduire les bons candidats des noms utilisés dans cette description leurs propriétés des adjectifs et autres qualifiants.

Identifier la sémantique des objets et des classes : on cherchera ensuite à identifier les actions que l'objet subit et provoque. Les verbes utilisés dans la description informelle de l'étape précédente fournissent les bons indices pour l'identification des opérations. C'est également à cette étape que l'on pourra définir les conditions d'ordonnancement des opérations, si nécessaire.

Identifier les relations entre les objets et les classes : l'objet étant maintenant identifié par ses caractéristiques et ses opérations, on définira ses relations avec les autres objets. On établira quels objets le "voient" et quels objets "sont vus" par lui.

Implanter les classes et les objets : la dernière étape consiste, bien entendu à implanter les objets en écrivant le code correspondant aux spécifications dans un langage de programmation. Lors de cette étape, on identifiera de nouveaux objets de plus bas niveau d'abstraction, ce qui provoquera l'itération de la méthode.

La méthode OOD répond à chaque question par des diagrammes données : le digramme de classes, le diagramme d'objets, le diagramme de modules et le diagrammes de processus. Pour comprendre le comportement dynamique des instances d'une classe, on associe à cette dernière un diagramme de transition d'états. Les diagrammes d'objets sont statiques, ils ne montrent pas le flot de contrôle, ni l'ordre des événements, pour ces raisons, la méthode fournit un diagramme chronologique qui représente le temps selon l'axe horizontal et les objets selon l'axe vertical.

Outils de modélisation :

La méthode de G. Booch, très complète, offre des outils permettant de couvrir tous les besoins possibles en matière de conception de systèmes informatiques de tous types. Il n'est néanmoins pas toujours nécessaire d'avoir recours à la totalité de ces outils dans le cadre d'un cas précis. C'est pourquoi nous n'allons présenter ici que les concepts qui nous paraissent les plus intéressants dans le cadre de la modélisation d'une base de données objets.

Il n'est pas possible, sauf pour des cas d'écoles vraiment simples, de représenter de façon unique, c'est-à-dire à l'aide d'un seul diagramme, toute la complexité d'un logiciel. Il est en général nécessaire d'aborder la conception selon plusieurs vues différentes. G. Booch a identifié quatre points de vue fondamentaux qui sont complémentaires deux à deux :

- Une vue Logique / Physique.
- Une vue Statique / Dynamique.

1) Modèles Logiques et Physiques : ils permettent de faire la distinction entre ce qui relève de l'analyse du problème (modèle logique) et ce qui relève de son implantation informatique (modèle physique). On utilisera pour cela quatre types de diagrammes :

- Les diagrammes de Classes.
- Les diagrammes d'Objets.
- Les diagrammes de Modules.
- Les diagrammes de processus.

2) Modèles Statiques et Dynamiques : les quatre types de diagrammes précédents décrivent essentiellement la structure statique du problème. Pour en décrire l'aspect dynamique c'est-à-dire, comment les objets sont crées puis détruits ? Comment ils s'échangent des messages ? On utilise deux autres types de diagrammes :

- Les diagrammes de Transition d'états.
- Les diagrammes de temps.

Les points faibles de cette méthode :

- OOD ne fournit aucune information sur la manière de composer les objets (agrégation, construction de types composés, ...),
- L'absence d'un langage de spécification et de conception,
- Les systèmes de bases de données ne sont pas bien supportés par la méthode.

1.4.3. La méthode HOOD (Hierarchical Object Oriented Design) :

HOOD est une méthode de conception orientée objet descendante développée pour le développement des logiciels « temps réels ». Elle a été conçue par l'agence CISI (Centre International de Service Informatique) de Toulouse (France) avec la collaboration des sociétés RATRA Espace après l'appel d'offre de l'Agence Spatiale Européenne (ASE) pour le projet « COLOMBUS ». La méthode permet une transition aisée vers le codage par génération de squelettes de code dans le langage cible. Elle est basée sur des formalismes graphique et textuel:

- Le formalisme graphique permet l'expression de la structure statique d'une solution de manière claire et accessible aux non spécialistes.
- Le formalisme textuel permet de compléter cette solution graphique par un squelette de description d'objets (ODS).

Les principes utilisés dans la méthode HOOD :

1) Principes d'abstraction, masquage de l'information et encapsulation : les objets sont définis seulement par leurs propriétés externes, leur interface) tandis que leur structure interne est cachée à l'utilisateur.

2) Principes hiérarchiques : les objets peuvent être décomposés en d'autres objets, c'est la relation « Parent-Enfant ». Un enfant n'existe pas de manière autonome dans le système : il n'est visible et donc utilisable que par ses frères. Les objets peuvent utiliser les opérations (les services) d'autres objets, c'est la relation « Senior-Junior ».

3) Principes de structuration du contrôle : les opérations sur les objets sont activées à travers les flots de contrôle.

La démarche de la méthode HOOD :

La démarche suivie dans la conception du modèle de HOOD, consiste à :

1) identifier les objets et les classes,
2) identifier les opérations et les relations,
3) relier les opérations et les relations aux objets,
4) décrire le formalisme graphique de HOOD.

Malgré que cette méthode ait été choisie dans plusieurs projets industriels comme méthode de conception, elle souffre encore de quelques limites :

- L'héritage n'existe pas dans HOOD, ce qui l'exclut des applications à décrire en langages à objets,
- La méthode de HOOD est attachée aux langages procéduraux, et plus particulièrement à Ada,
- Elle est adaptée à la conception des logiciels temps réel programmés en Ada bien que son langage graphique ne permette pas de représenter les différents modes de synchronisation,
- Elle ne compte pas de langage de spécification pour une vérification de cohérence de la méthode.

1.4.4. OMT (Object Modeling Technique) :

Introduction :

La méthode OMT a été inventée dans le centre de recherche et de développement de *General Electrique* à la fin des années 80. Le principal ouvrage décrivant la méthode est paru en 1991 **[RUM 91]**.

Les modèles de la méthode :

La méthode fournit trois modèles principaux pour décrire les aspects statique, dynamique et fonctionnel. Le modèle statique est une extension du modèle Entité-Association vers les concepts des objets (classe, héritage, opération, agrégation, généralisation). Le modèle dynamique est basé sur les diagrammes d'états/transitions avec spécification des événements qui déclenchent les transitions, des opérations de transformation et des attributs concernés par ces transformations. Le modèle fonctionnel est basé sur les diagrammes de flux de données (DFD) classiques. Ces diagrammes sont utilisés pour spécifier des fonctions qui vont opérer sur des objets. Chaque processus élémentaire d'un diagramme de flux fait référence à une opération définie sur un objet.

La démarche méthodologique :

OMT suit les quatre étapes suivantes :

a) *L'étape d'analyse :* permet d'élaborer les trois modèles conceptuels (Statique, dynamique et fonctionnel). Elle permet de décrire ce que doit faire le système et non du « comment » le faire.

b) *L'étape de conception du système :* permet de définir l'architecture, les constituants et les ressources du système informatique. Elle permet de décomposer le système d'information en sous systèmes, d'identifier les éléments de conflits, de choisir la stratégie de stockage et d'accès aux données et d'implémenter le contrôle avec ses contraintes et ses exceptions.

c) *L'étape de conception des objets :* est une spécification détaillée de l'implémentation des objets, indépendante d'un environnement donné, mais compatible avec une technologie choisie, c'est le niveau logique. C'est à ce niveau que l'ensemble des opérations sont complètement identifiées et spécifiées (avec leurs algorithmes).

d) *l'étape d'implémentation :* c'est le niveau physique et indépendant de la technologie matérielle choisie.

Comme points forts de la méthode, on peut citer :

- qu'elle offre un modèle objet riche,
- qu'elle possède une représentation graphique très conviviale avec un choix judicieux des symboles utilisés.

Cependant OMT souffre de quelques points :

- il n'est pas clair si les attributs peuvent ou non avoir un type complexe,
- l'identification des objets par leurs attributs gène l'évolutivité des structures.

Les méthodes HOOD, OMT, OOD sont des méthodes Orientées Objets, qui utilisent un langage de spécification graphique UML qu'on va présenter dans la section suivante :

La Notion Unifiée UML (Unified Modeling Language) :

Généralités :

UML (Unified Modeling Language) est un langage graphique de modélisation objet, résultat des tentatives de l'unification des méthodes OOD de Booch, OMT de Rumbangh et OOSE de Jacobson.

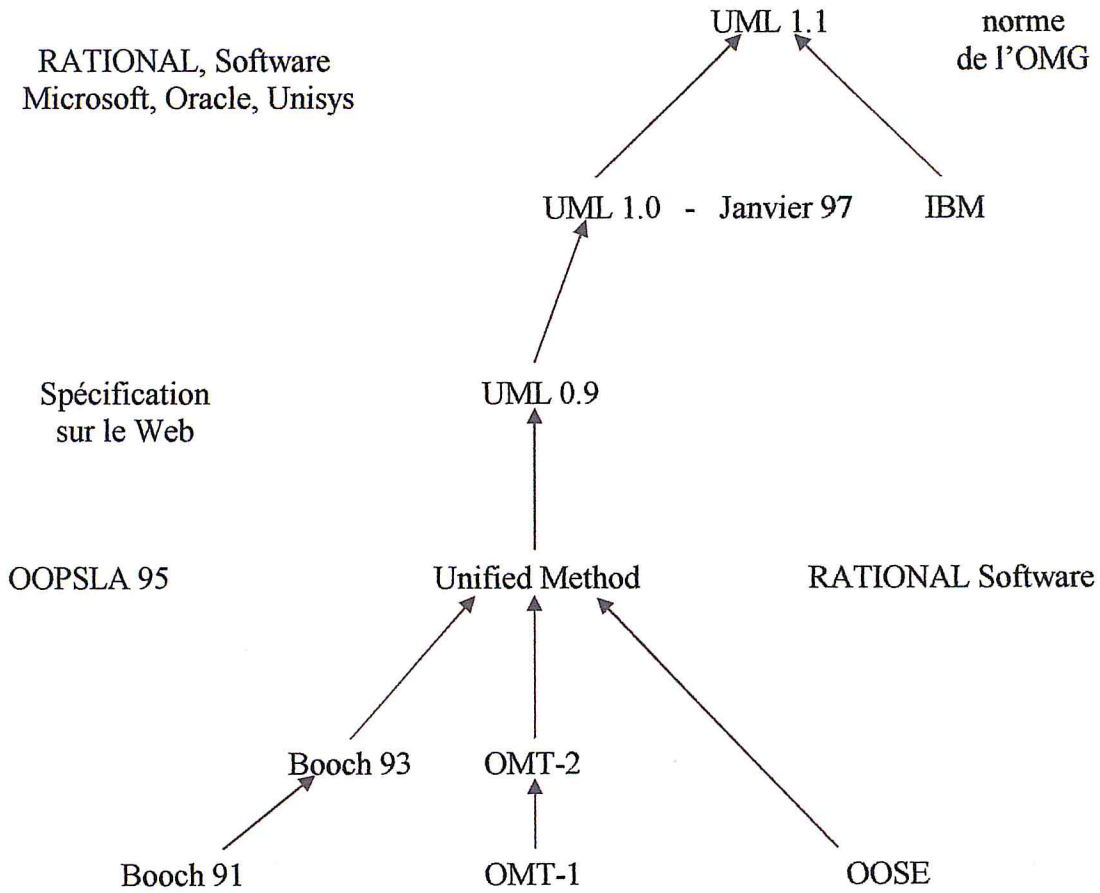
- Le rapprochement en premier d'OMT et de OOD a permis :

1. de garder la puissance d'expression de la sémantique d'OMT à travers les concepts d'association entre objets et l'expression de leur comportement à travers la notion de diagramme d'état / transition.
2. de sélectionner de OOD les concepts de modules (sous-systèmes) et la notion de flots de messages.

- Le rapprochement en second lieu avec OOSE a permis : d'utiliser les « Use Case » (cas d'utilisation) comme moyen d'expression des besoins des utilisateurs dans la phase d'analyse.

A défaut de trouver un consensus pour la méthode unifiée, les trois chercheurs s'entendent pour la définition d'une notion universelle UML qui devient une norme de l'OMG (Object Management Group) en Novembre 1997. UML est une méthode de notation graphique permettant de représenter les modèles objets en phase d'analyse mais aussi une façon standard de les stocker.

Historique d'UML :



Présentation des concepts de la notion unifiée:

UML est constituée de neuf diagrammes, comme le montre la figure suivante :

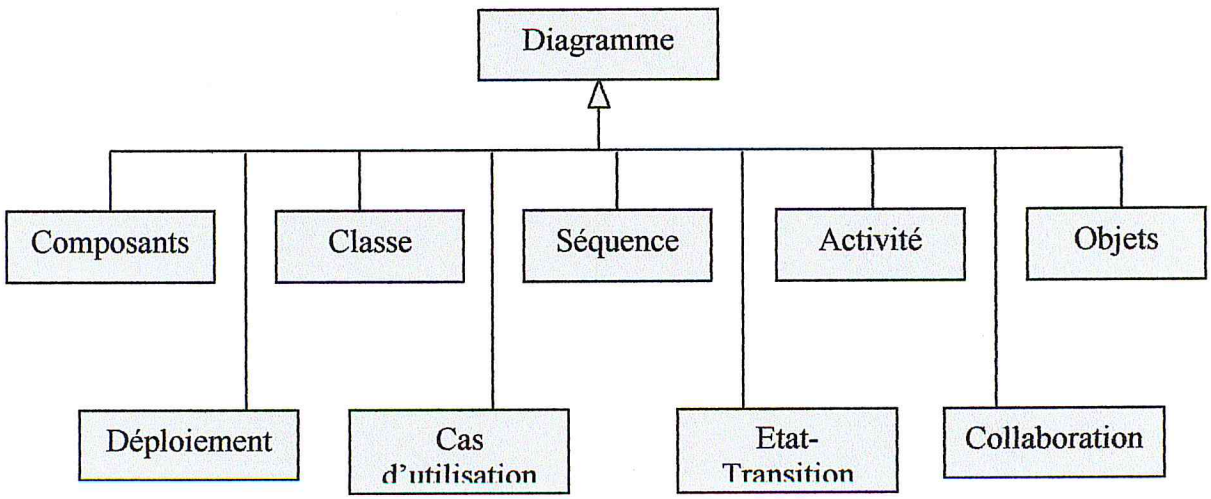


Figure IV.1.1 : *Les diagrammes de la notion unifiée.*

- Les diagrammes de classe et d'objets permettent de représenter l'aspect statique des objets.
- La représentation des interactions entre les objets est représentée par les diagrammes de séquence et de collaboration appelés encore diagrammes d'interaction.
- Le comportement des objets et la description des opérations sont représentés par les diagrammes d'états-transitions et des diagrammes d'activité.
- La représentation de la partie physique et des relations entre logiciel et matériel sont décrites par les diagrammes de composants et de déploiement.
- Enfin les diagrammes de cas d'utilisation permettent de mieux enregistrer les besoins des clients et apportent une aide tout au long du processus de développement.

1) Les diagrammes de classe et d'objets :

Les diagrammes de classe décrivent les types d'objets (classes) et leurs relations. Les objets sont les instances de classe et les liens sont instances de relations. Deux catégories de liens sont définies : les associations et les liens de sous-typage.

2) Les diagrammes d'interaction :

Les diagrammes d'interaction sont constitués de deux diagrammes : les diagrammes de collaboration et les diagrammes de séquence.

a- Les diagrammes de collaboration :

Ils sont associés aux digrammes d'objets et représentent les interactions entre les objets. Une interaction est réalisée par un groupe d'objet qui collaborent en échangeant des messages. L'interaction est représentée par un arc étiqueté (le message) reliant deux diagrammes d'objets.

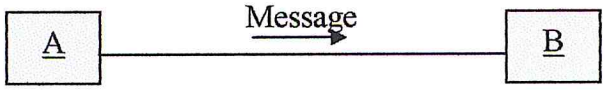


Figure IV.1.2 : *Représentation graphique du diagramme de collaboration.*

b- Les diagrammes de séquence :

Ils représentant l'interaction entre les objets en insistant sur la chronologie des envois de messages.

- Messages : flèches horizontales, allant de l'objet émetteur vers l'objet récepteur. Un axe vertical permet de positionner les messages dans le temps.
- Période d'activité de l'objet : bande rectangulaire placée sur l'axe temporel.

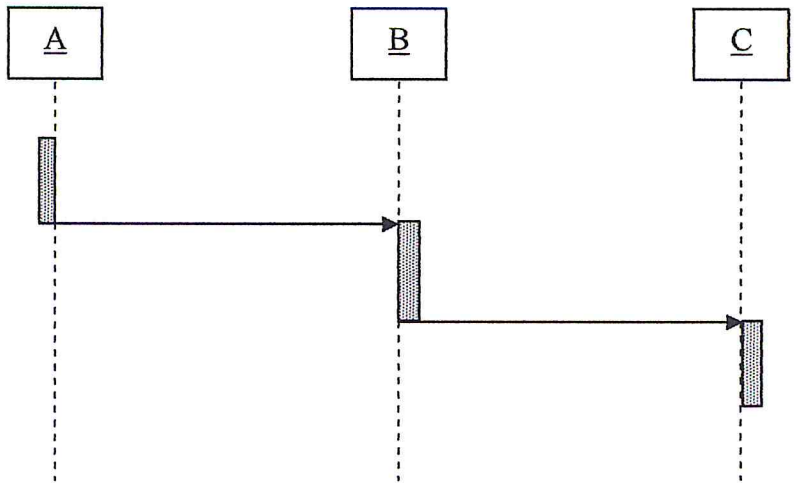


Figure IV.1.3 : *Représentation du diagramme de séquence.*

3) Les diagrammes d'états-transitions et diagrammes d'activités :

a- Les diagrammes d'états-transitions :

Le digramme d'états-transitions décrit le comportement exhaustif de chaque objet. De manière informelle, le comportement d'un objet peut être décrit en termes d'états d'événements.

Le lien qui existe entre les opérations d'écritures dans le diagramme de classes et les diagrammes d'états-transitions et diagrammes d'activités s'exprime par les notions d'action et d'activité. L'action est instantanée, atomique et associée à la transition. L'activité a une durée et est associée à un état.

Ainsi, événement, transitions et états sont indissociables dans la description du comportement des objets.

Le diagramme d'états-transitions est un graphe dirigé qui relie les concepts : événement, transition et état. Chaque transition peut être précisée par l'action associée à l'événement.

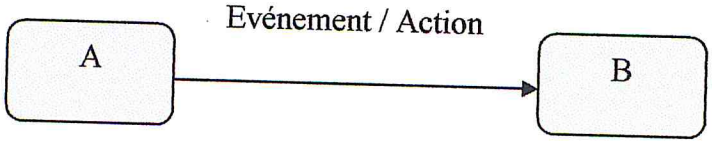


Figure IV.1.4 : *Le diagramme d'états-transitions.*

b- Le diagramme d'activité :

Il est une variante des digrammes d'états-transitions et exprime surtout le comportement interne des méthodes. Il est basé sur le concept d'activité qui représente une étape dans l'exécution de la méthode. Ainsi, lorsqu'une activité se termine, la transition se déclenche et l'activité suivante démarre. Les activités peuvent être synchroniseés au moyen de barre de synchronisation. La représentation graphique suivante est associée :

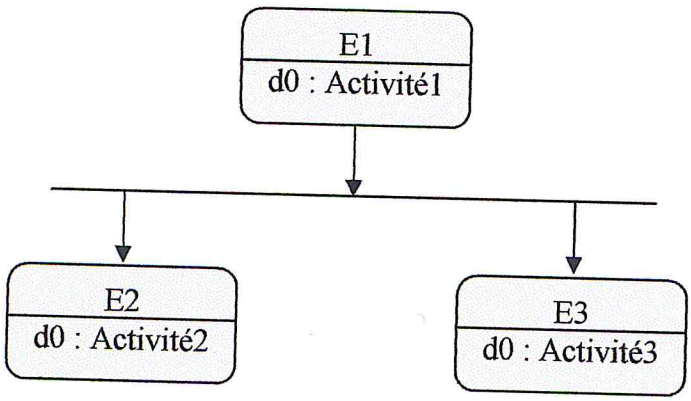


Figure IV.1.5 : *Le diagramme d'activité.*

4) Les diagrammes de composants et de déploiement :

a- Les diagrammes de composants :

Les diagrammes de composants décrivent les éléments physiques. Chaque classe du modèle est réalisée à l'aide de deux composants : la spécification et le corps. La spécification contient l'interface de la classe alors que le corps contient l'implémentation.

Les composants peuvent être reliés par des liens de dépendance qui expriment qu'un composant peut utiliser un autre composant.

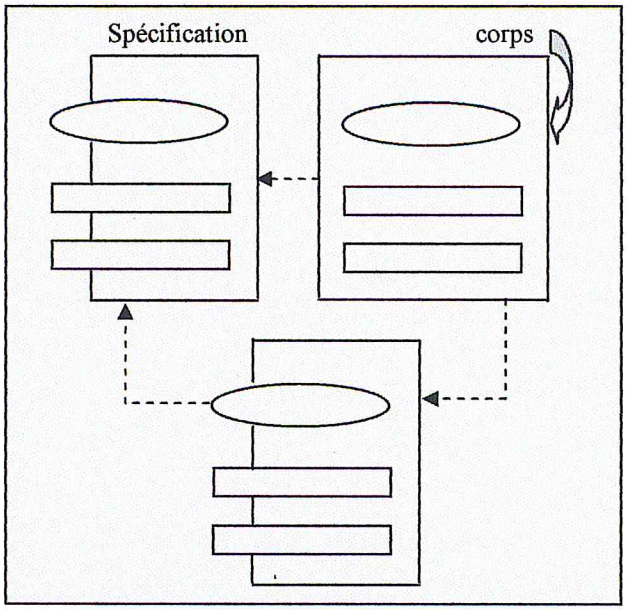


Figure IV.1.6 : *Le diagramme de composants.*

b- Les diagrammes de déploiement :

Ils montrent la disposition physique des différents matériels, ainsi que la répartition des programmes exécutables sur ces matériels. Chaque ressource est représentée par un cube qui contient un nom identifiant le matériel, les classes ou les objets.

Les cas d'utilisation :

Les cas d'utilisation sur UML sont repris des cas d'utilisation de Jacobson. Le modèle de cas d'utilisation comprend les acteurs, le système et le cas d'utilisation lui-même.

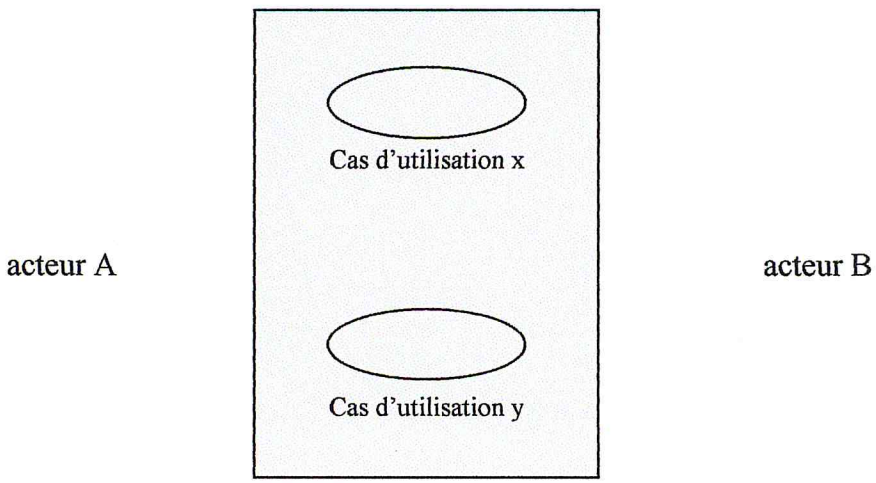


Figure IV.1.7 : *Les cas d'utilisation.*

Nous venons de représenter succinctement la notion UML. Nous rappelons que UML n'est qu'un langage de spécification et non pas une méthode, car rien n'est dit sur comment utiliser ces diagrammes.

IV.2. APPROCHE PRÉCONISÉE

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. Son cycle de développement couvre les étapes de spécification formelle (modélisation conceptuelle) jusqu'à l'implémentation (conception physique).

Elle offre un modèle objet riche et une représentation graphique très conviviale.

Elle allie judicieusement l'approche objet avec l'approche fonctionnelle plus classique. 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. On précisera que les extensions apportées sont l'agrégation et la généralisation d'une part, et la spécification d'opérations sur les objets d'autre part, un nombre de contraintes est spécifié également dans le modèle objet.

IV.3. PRÉSENTATION DE LA MÉTHODE OMT (Object Modeling Technique) :

Le modèle conceptuel de OMT est constitué, comme indiqué déjà, de trois modèles: le modèle objet, le modèle dynamique et le modèle fonctionnel.

3.1. Le modèle objet

Le modèle objet décrit la structure statique d'un système en montrant les objets, leurs identités, leurs relations avec les autres objets, ainsi que leurs attributs et leurs opérations. Il décrit les structures de données sur lesquelles les modèles dynamique et fonctionnel opèrent. Il est représenté graphiquement par des diagrammes d'objets contenant des classes organisées en hiérarchie.

3.1.1. Objet et classe

a) Objet

Un objet représente une entité du monde réel qui se caractérise par une identité. Des états significatifs et un comportement.

- L'identité d'un objet est la propriété qui permet de distinguer chaque objet par rapport aux autres.
- L'état d'un objet correspond aux valeurs de tous ces attributs à un instant donné. Les propriétés sont définies dans la classe d'appartenance de l'objet.
- Le comportement d'un objet est défini par l'ensemble des opérations qu'il peut exécuter en réaction aux messages envoyés par les autres objets. Il se décrit en terme de contraintes et d'opérations.

Les contraintes définissent les états possibles ainsi que les changements d'états de cet objet.

Les opérations permettent de manipuler l'état de cet objet. Les opérations sont définies dans la classe d'appartenance de l'objet.

b) Classe

Une classe d'objet décrit un groupe d'objets ayant des propriétés similaires (attributs), un comportement commun (opérations) et des relations communes avec les autres objets.

Les objets et les classes sont modélisés respectivement par les diagrammes d'objets (diagramme d'instances) et les digrammes de classes. La figure ci-dessous montre un diagramme des classes (à gauche) et un diagramme d'instances (à droite). Les objets X, Y sont des instances de la classe employé.

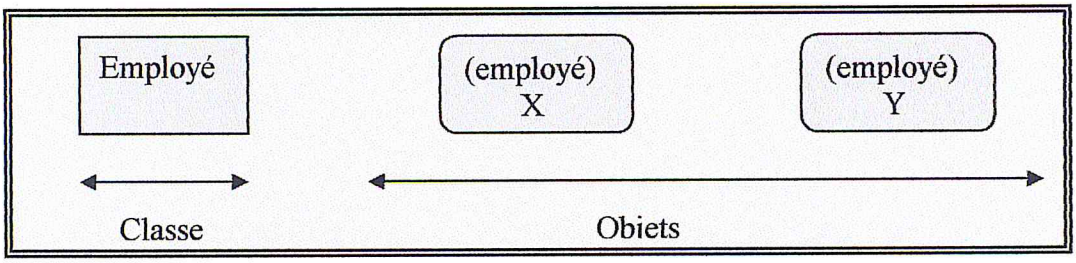


***Figure IV.2.1** : Classes et objets.*

c) Attributs

Un attribut est une donnée (information d'état) pour laquelle chaque objet dans une classe a une valeur propre.

d) Les opérations et les méthodes

Une opération correspond à une action ou transformation applicable sur les objets ou par les objets d'une classe ; « Supprimer», « Imprimer» sont des opérations de la classe Fichier. Tous les objets d'une classe partagent les mêmes opérations. Une même opération peut s'appliquer sur plusieurs classes d'objets, elle est donc polymorphe.

Une méthode est l'implémentation d'une opération dans une classe. La figure *IV.2.2* résume la représentation graphique d'une classe. Les informations qui suivent les noms d'attributs et d'opérations sont facultatives.

Nom-de-Classe
Nom-d'attribut-1 : type-de-donnée-1 = valeur-par-defaut-1 Nom-d'attribut-2 : type-de-donnée-2 = valeur-par-defaut-2 . . .
Nom-d'opération-1(liste-d'argument-1) : typt-de-résultat-1 Nom-d'opération-2(liste-d'argument-2) : typt-de-résultat-2 . . .

***Figure IV.2.2 :** Résumé de la notation de modélisation objet pour les classes.*

3.1.2. Liens et associations

Les liens et les associations permettent d'établir des relations entre objets et classes. Un lien est une collection entre des instances d'objets, par exemple ; Mohamed Propriétaire de Toyota.

Une association décrit un groupe de liens ayant une structure sémantique commune, par exemple « Propriétaire de» est une association de la classe *Personne* avec la classe *Voiture.*

Une association peut être binaire, ternaire ou n-aires.

La figure *IV.2.3* montre une association et les liens correspondants. L'association (ou le lien) est représentée par une ligne joignant respectivement les classes et les objets et est nommée par un verbe écrit en italique sur la ligne.

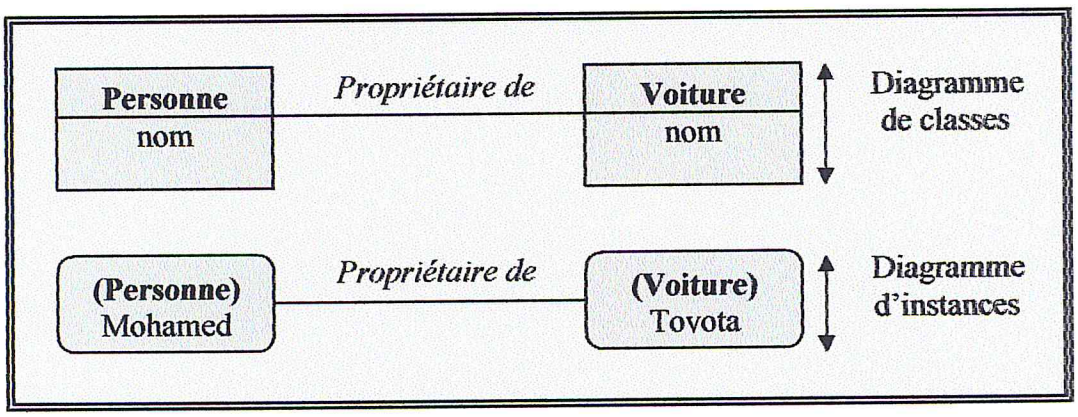


***Figure IV.2.3** : Association un-à-un et lien.*

Les associations ternaires sont modélisées par un losange avec des lignes les reliant aux classes concernées. La figure *IV.2.4* montre une association ternaire : des étudiants sont inscrits chaque année à des modules.

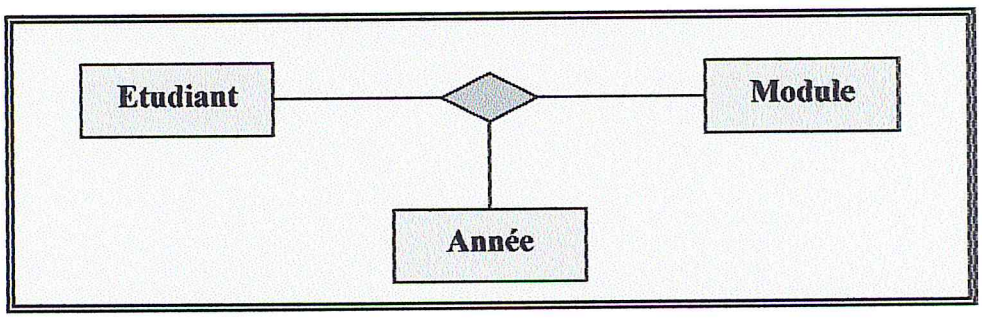
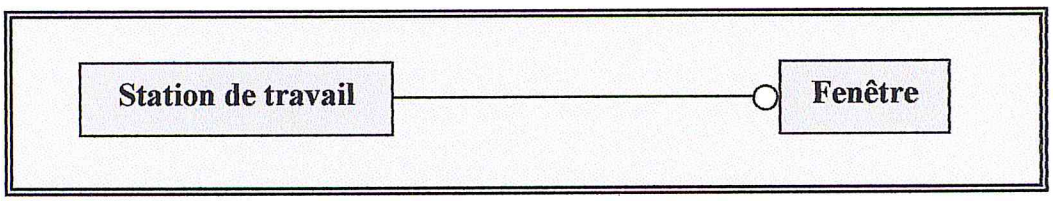


***Figure IV.2.4** : Association ternaire.*

Pour modéliser les associations, des constructions additionnelles sont utilisées :

a) La multiplicité

La multiplicité indique le nombre d'instances d'une classe qui peuvent êtres liés à une instance de la classe. On la décrit souvent comme étant "un" ou " plusieurs". Le cercle blanc (—○) est le symbole de zéro ou un, le cercle noir (—●) est le symbole pour "plusieurs", signifiant un ou plus. Une ligne sans symbole de multiplicité (—) indique une association qui a une multiplicité à 1 de chaque côté.



FigureIV.2.5 : *Multiplicité zéro -ou - un.*

b) Attributs de liens

Les attributs de liens sont des propriétés de liens d'une association. Chaque attribut de lien possède une valeur pour chaque lien. La notation pour un attribut est une boite attachée à l'association par une boucle. Un ou plusieurs attributs de liens peuvent apparaître dans la deuxième partie de la boite. Dans la figure *IV.2.6*
« Séance, Salle » sont des attributs de liens de l'association << Enseigne >>.

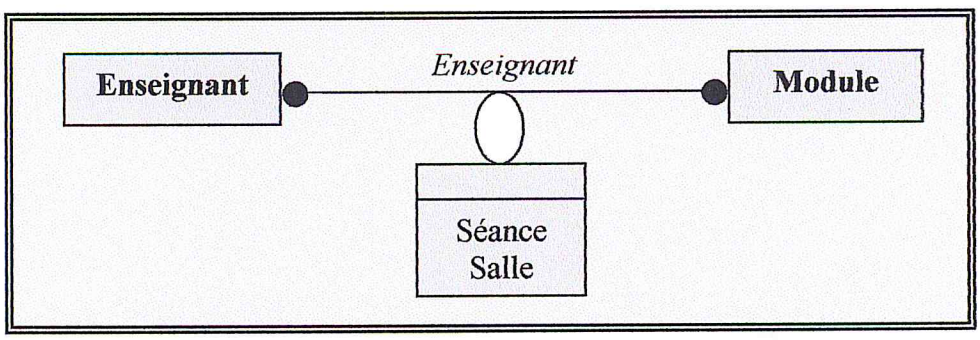


Figure IV.2.6 : *Attributs de lien pour une association plusieurs à plusieurs.*

c) Modélisation d'une association en classe

Il est parfois utile de modéliser une association comme une classe quand les liens peuvent participer à des associations avec d'autres objets ou quand les liens sont sujets à des opérations. Dans ce cas, chaque lien devient une instance de la classe. La figure *IV.2.7* montre les informations nécessaires pour enseigner un module. Les enseignants peuvent être chargés de plusieurs modules, ils possèdent un emploi du temps pour les modules qu'ils enseignent.

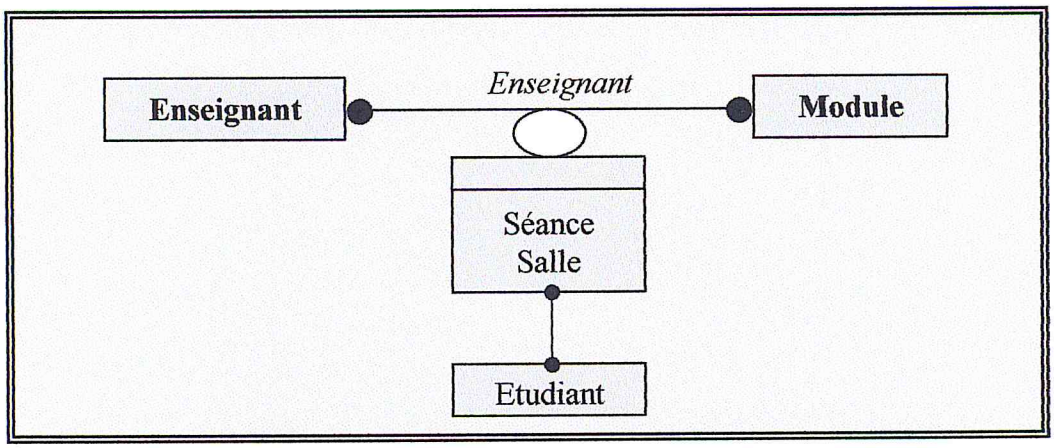


Figure IV.2.7: *Modélisation d'une association en classe.*

d) Les noms de rôle

Un rôle est une extrémité de l'association. Une association binaire a deux rôles, chacun d'eux pouvant avoir un nom de rôle. Ce nom de rôle identifie de façon unique une extrémité de l'association (Figure *IV.2.8*).

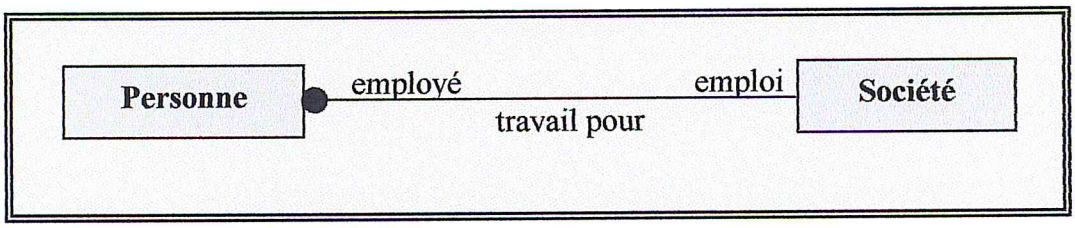


Figure IV.2.8 *: Nom de rôle.*

e) Qualification

Une association qualifiée met en relation deux classes d'objets et un qualificatif. Le qualificatif est un attribut spécial qui réduit la multiplicité effective d'une association. Il est

dessiné comme une petite boite prés de la classe qu'il qualifie. Dans la figure ci dessous, «Module» et «Enseignant» sont des classes d'objets et Nom enseignant est un qualificatif.

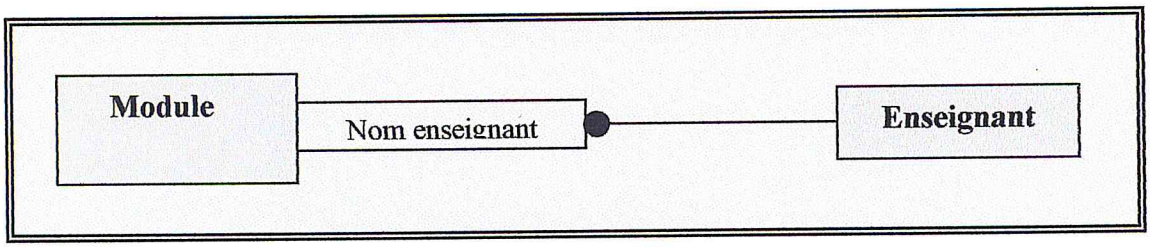


Figure IV.2.9 : *Association qualifiée.*

3.1.3. L'agrégation

L'agrégation est une forme spéciale de l'association, c'est une relation «composé-composant» ou «partie-de» dans laquelle les objets représentant les composants d'une chose sont associés à un objet représentant l'assemblage entier. L'agrégation est dessinée comme une association, à l'exception d'un petit losange qui indique l'extrémité d'assemblage de la relation. La figure ci-dessous montre le cas d'une société composée de plusieurs divisions, qui elles-mêmes se décomposent en services.

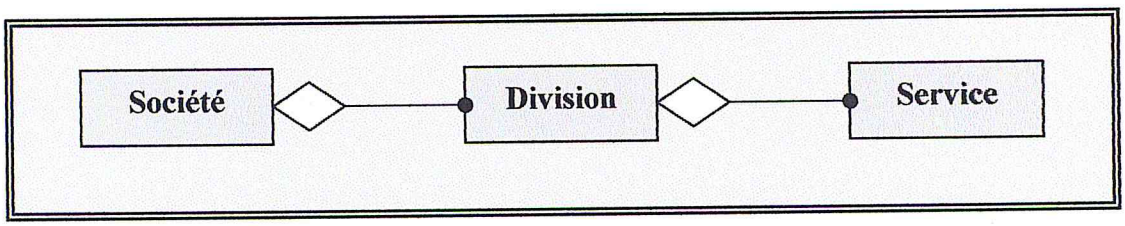


Figure IV.2.10 : *Agrégation.*

Les opérations sur un agrégat se propagent souvent sur les composants. La figure *IV.2.11* montre un exemple de propagation. L'opération «Créer» se propage de la société aux divisions, puis aux services. L'opération ne se propage pas à rebours. La propagation est indiquée par une flèche et un nom d'opération prés de l'association affectée. La flèche indique la direction de la propagation.

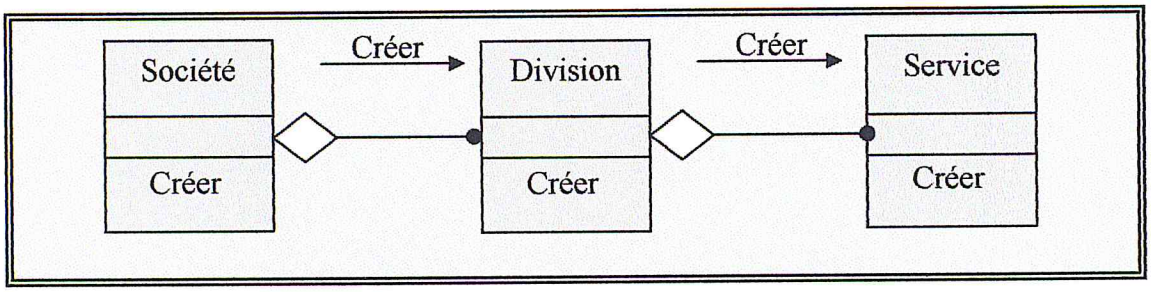


Figure IV.2.11 : *La propagation des opérations.*

3.1.4. Généralisation et héritage

La généralisation est la relation entre une classe appelée « super-classe » et une ou plusieurs versions affinées de la classe appelée `sous-classe'. Chaque sous-classe hérite les propriétés de sa super-classe de plus, une sous-classe possède ses propres attributs et opérations spécifiques. Une sous-classe peut redéfinir une caractéristique d'une super-classe en définissant une propriété portant le même nom. La propriété redéfinie de la sous-classe affine et remplace la propriété de la super-classe, on peut redéfinir les valeurs par défaut des attributs, et les méthodes d'opérations. Mais on ne doit jamais redéfinir la signature ou forme d'une propriété. La figure *IV.2.12* illustre le concept d'héritage. La notation pour la généralisation est un triangle reliant une super-classe à une sous-classe. Pour l'héritage multiple ; le triangle vide indique que les sous-classes sont disjointes, alors que le triangle plein indique que les sous-classes se recouvrent.

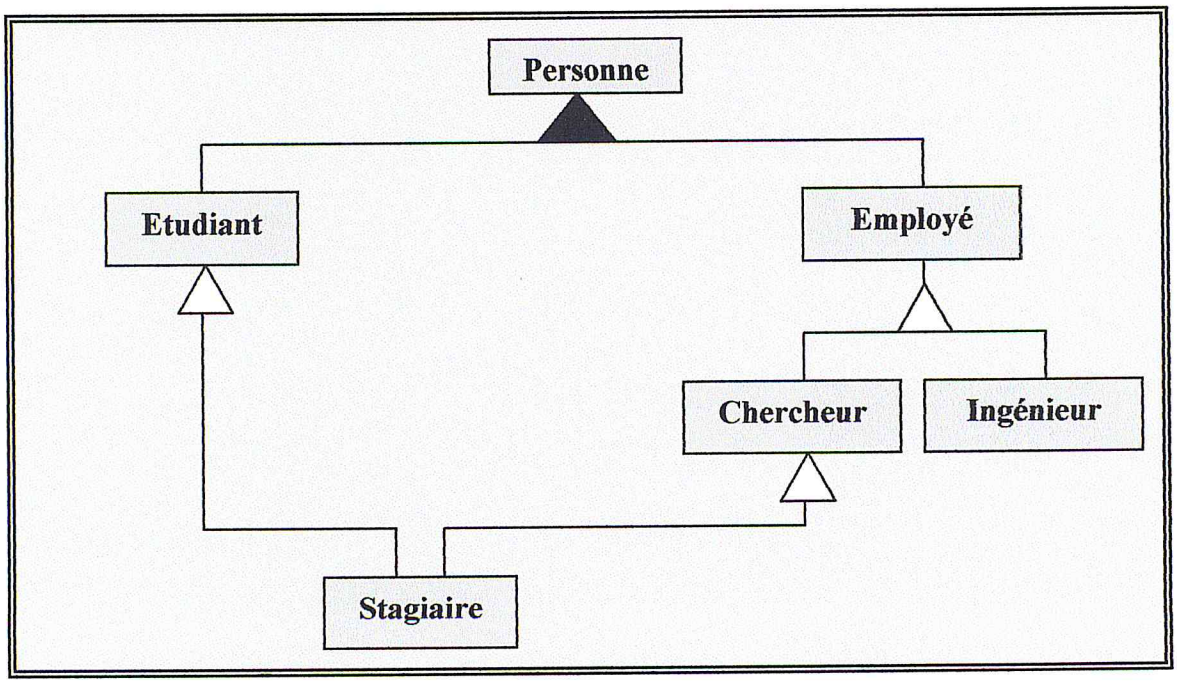


Figure IV.2.12 : *Héritage simple et multiple.*

3.1.5. Regroupement des constructions

Il s'agit de diviser le système d'information en feuillets de taille uniforme, ce qui le rend claire pour les perspectives de dessin, de compréhension, etc...

a) Les feuillets

Un modèle complexe ne pourra pas être représenté sur un seul morceau de papier. Le feuillet est le mécanisme de partitionnement d'un gros modèle objet en série de pages. Chaque feuillet a un titre et un numéro. Chaque association et chaque généralisation apparaissent sur un feuillet unique. Des classes peuvent apparaître sur plusieurs feuillets des copies multiples de la même classe forme un point de connexion entre feuillets.

b) Les modules

Un module est un ensemble de plusieurs feuillets représentant une vue logique d'un sous ensemble du modèle.

3.2. Le modèle dynamique

L'objectif de ce modèle est de décrire le cycle de vie des objets (listes d'états possibles). Pour cela, OMT utilise les diagrammes d'états/transitions sur lesquels sont portés les événements déclencheurs des transitions et les opérations de transformation correspondantes. Le passage d'un état à un autre se traduit par la modification de la valeur d'un ou plusieurs attributs de l'objet. Une transition est décrite par l'événement qui la déclenche, les attributs qui caractérisent cet événement, la condition a vérifié en plus de l'occurrence d'événement et l'action à exécuter.

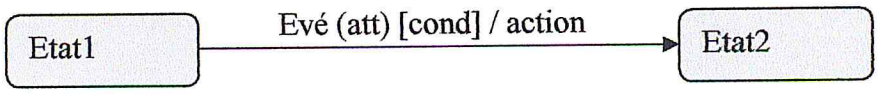


Figure IV.2.13 : *Diagramme d'états / transitions.*

3.2.1. L'événement

Un événement est un moyen de transition d'information d'un objet vers l'autre, il peut donc comporter des attributs. Un événement est sans durée, mais l'attribut date le caractérise implicitement. Les événements sont groupés en classes.
Ils peuvent s'enchaîner par des liens de causalité ou survenir de façon concurrente. La figure *IV.2.14* montre des exemples d'événement avec des attributs.

Chiffre composé (chiffre). Insérer carte (carte). Combiné soulevé. Température maximale.

Figure IV.2.14 : *Classe et attributs d'événements.*

3.2.2. Les scénarios et suivi d'événements [BOU 94]

Pour aider à identifier les événements et à réaliser les digrammes d'états des classes, OMT propose une représentation intermédiaire, appelée scénario. Un scénario est une séquence d'événements qui arrivent durant une exécution particulière du système. Il peut être considéré comme une trace historique d'une exécution ou d'une simulation du système. Un système peut être défini par plusieurs scénarios, correspondant à des vues particulières du fonctionnement du système. La figure *IV.2.15* montre le scénario de la gestion des transactions bancaires.

L'utilisateur insère la carte Demander mot de passe Entrer mot de passe Vérifier compte Compte banque correcte Demander montant Entrer montant Traiter transaction Transaction correcte Délivrer billets Billets pris Terminer Imprimer reçu Ejecter carte

Figure IV.2.15 : *Scénario pour la gestion des transactions bancaires.*

L'ensemble des scénarios constitue l'activité du système d'information. Ces scénarios représentent en quelque sorte les fonctions de système. La synthèse de l'ensemble des scénarios permet d'identifier, pour chaque classe d'objets, les événements qui arrivent ou qui sont émis par cette classe. C'est à partir de ces événements arrivant ou partant que l'on peut élaborer le diagramme d'états d'une classe (Figure *IV.2.16*).

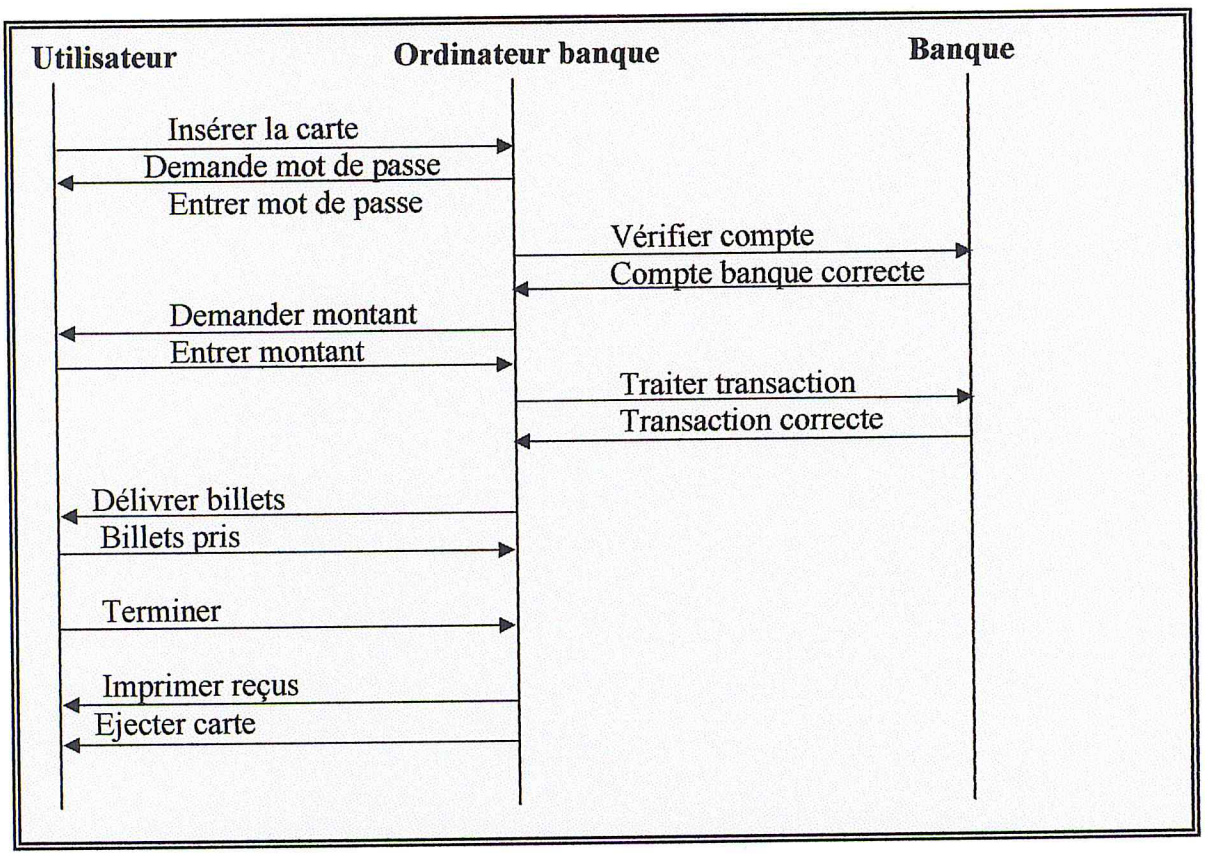


Figure IV.2.16 : *Diagramme de suivi d'événement pour la gestion des transactions bancaires.*

3.2.3. Transition

Une transition est définie comme l'occurrence d'un événement et l'effet correspondant, c'est-à-dire l'action à exécuter. Cette action est considérée comme atomique.

3.2.4. L'état [BOU 94]

Un état est la valeur d'un objet durant un intervalle de temps ou durant l'occurrence de deux événements. Durant cet intervalle, un objet peut réaliser une activité qui ne change pas son

état (ou que les changements réalisés ne sont pas considérés comme des états remarquables à mémoriser). Cette activité peut être aussi considérée comme une activité de contrôle de l'état courant de l'objet. Cette activité peut être composée d'une ou plusieurs actions atomiques. L'activité d'un objet dans un certain état est représentée dans le nœud qui représente cet état et est introduite par le mot clé do (figure *IV.2.17*).

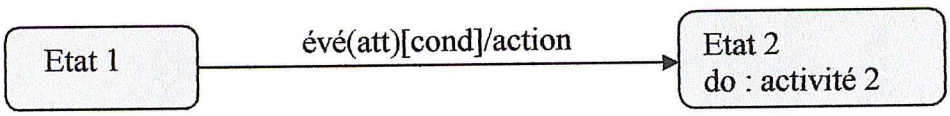


***Figure IV.2.17** : Diagramme d'états / transitions avec activité.*

3.2.4.1 Les conditions :

Une condition est une fonction booléenne des valeurs de l'objet telle que «la température est < 0 », et qui est valide pendant un intervalle de temps. Les conditions peuvent être des gardes sur les transitions ; une transition est franchie quand un événement survient mais quand la condition est vraie. La condition sur une transition est représentée entre crochets suivant le nom de l'événement.

3.2.4.2 Les opérations :

Un état doit pouvoir spécifier ce que fait un objet en réponse à un événement. On veut pouvoir donc effectuer une opération. Il existe deux types d'opérations : les activités et les actions.

Une activité : c'est une opération qui nécessite un certain temps d'exécution ; elle est associée à un état et est notée « do : A » à l'intérieur d'une boite d'état, cela indique que l'activité A commence en entrant dans l'état et se termine à sa sortie.

Une action : elle représente une opération dont la durée est insignifiante par rapport à la durée du diagramme d'états. Une action se note à la suite du nom de l'événement et est précédée par une barre oblique « / ».

3.2.5. Diagramme d'état [BOU 94]

Chaque classe d'objets, qui nécessite l'étude de son cycle de vie, sera représentée par un diagramme d'états. Les diagrammes d'états de l'ensemble des classes constituent le modèle dynamique du SI. Dans le diagramme ci-dessous, les principaux événements sont des clics souris réalisés par l'utilisateur. Les actions accompagnant chaque événement sont considérées comme atomiques. Les activités afficher et masquer introduites par le mot clé do sont des activités permanentes.

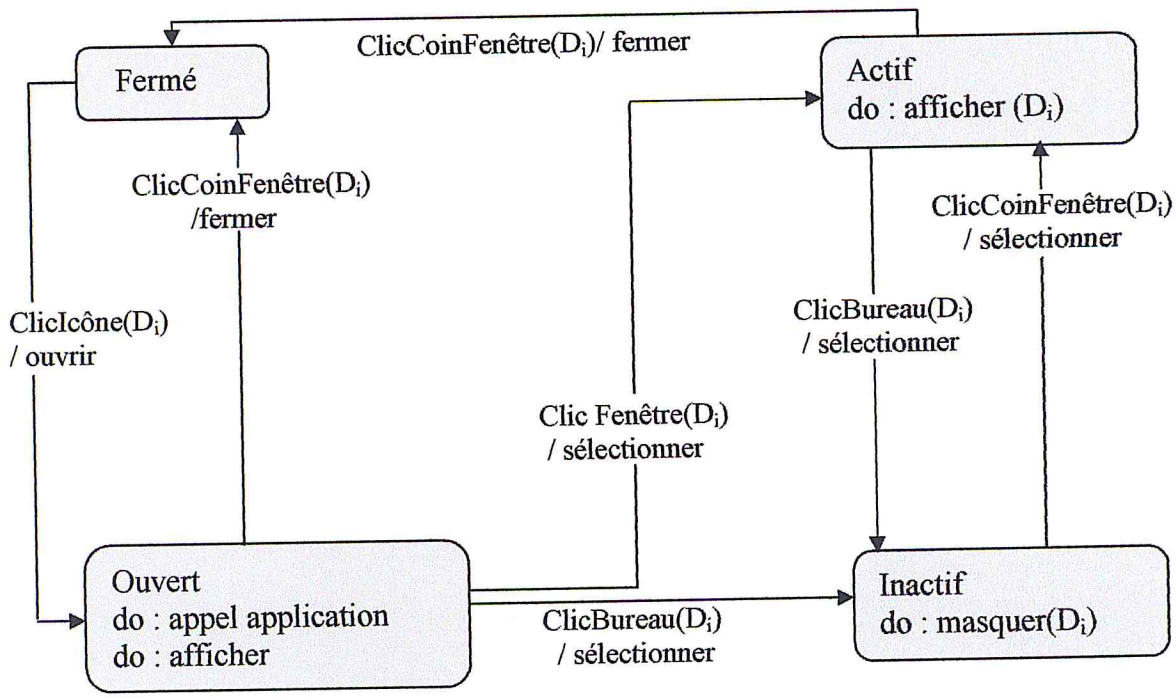


***Figure IV.2.18**: Exemple de modèle dynamique.*

3.3. Le modèle fonctionnel

Le modèle fonctionnel décrit les processus de transformation de l'application (ou les fonctions). Ce processus est une formalisation opérationnelle d'un scénario. OMT utilise le modèle fonctionnel pour décrire les fonctions du système d'information. Le modèle utilisé est basé sur les diagrammes de flux de données (DFD) classiques.

3.3.1. Les digrammes à flots de données

Un digramme à flots de données (DFD) montre les relations fonctionnelles entre les valeurs calculées par un système. C'est un graphe qui montre les flots de valeurs des données à partir de leur source dans les objets, en passant par les traitements qui les transforment, vers leur destination dans d'autres objets. Un digramme à flots de données comporte des traitements qui transforment les données, des flots de données qui transportent les données, des objets acteurs qui produisent et consomment les données et des objets réservoirs de données qui stockent passivement les données.

o Les traitements

Le traitement transforme les valeurs des données, il est représenté par une ellipse contenant une description de transformation.

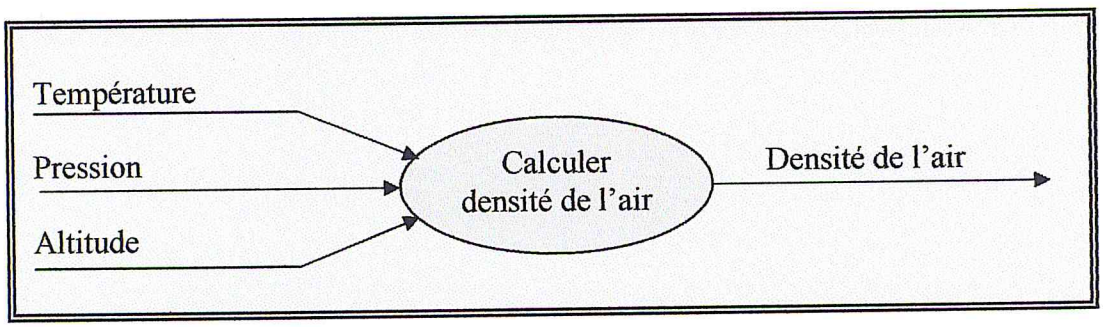


Figure IV.2.19 : *Traitement de calcul densité de l'air.*

o Les flots des données

Un flot des données relie la sortie d'un objet ou d'un traitement à l'entrée d'un autre objet ou traitement. On représente un flux de données par une flèche entre le producteur et le consommateur de la valeur de donnée. La flèche possède une étiquette portant la description des données.

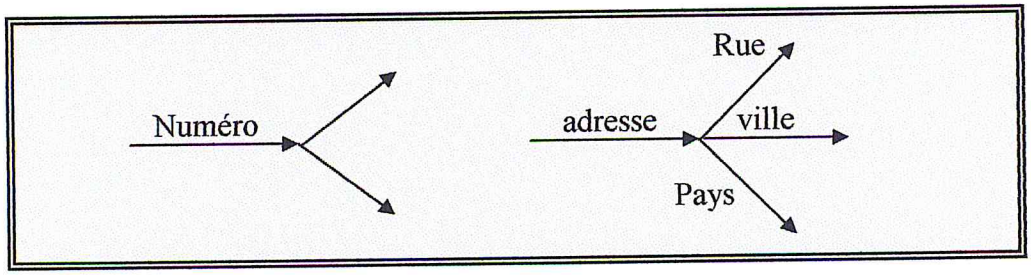


Figure IV.2.20 *: Flots de données pour copier une valeur et composer une valeur d'agrégat.*

o Les acteurs

Un acteur est un objet actif qui dirige le graphe de flot de données en produisant ou en consommant des valeurs. L'utilisateur d'un programme est un acteur. On représente un acteur par un rectangle pour montrer qu'il s'agit d'un objet. Les flèches entre l'acteur et le diagramme sont les entées et les sorties du diagramme.

o Les réservoirs de données

Un réservoir de données est un objet passif à l'intérieur d'un diagramme à flots de données qui stocke des données pour un accès ultérieur. A la différence d'un acteur, un réservoir de données n'engendre pas d'opérations par lui-même mais répond simplement à des requêtes pour stocker les données et à y accéder.

Le réservoir de données est représenté par une paire de lignes parallèles contenant le nom du réservoir. Les flèches entrantes indiquent les informations et les opérations qui modifient les données stockées : ajout d'éléments, modification de valeurs ou suppression d'éléments. Les flèches sortantes indiquent l'information récupérée à partir du réservoir. Il peut s'agir de la récupération de la valeur entière ou d'une de ces opérations.

La figure ci dessous montre un réservoir de données concernant un compte en banque. La flèche à double sens indique que solde est à la fois une entrée et une sortie de l'opération de retrait. On pourrait représenter cela par deux flèches séparées. Ainsi elle montre une liste de prix pour des articles. Une entrée de réservoir consiste en un nom et un coût. Si l'article est donné, alors le coût correspondant est trouvé. La flèche sans étiquette partant du réservoir de données et allant vers le traitement indique que la liste entière de prix est une entrée de l'opération de sélection.

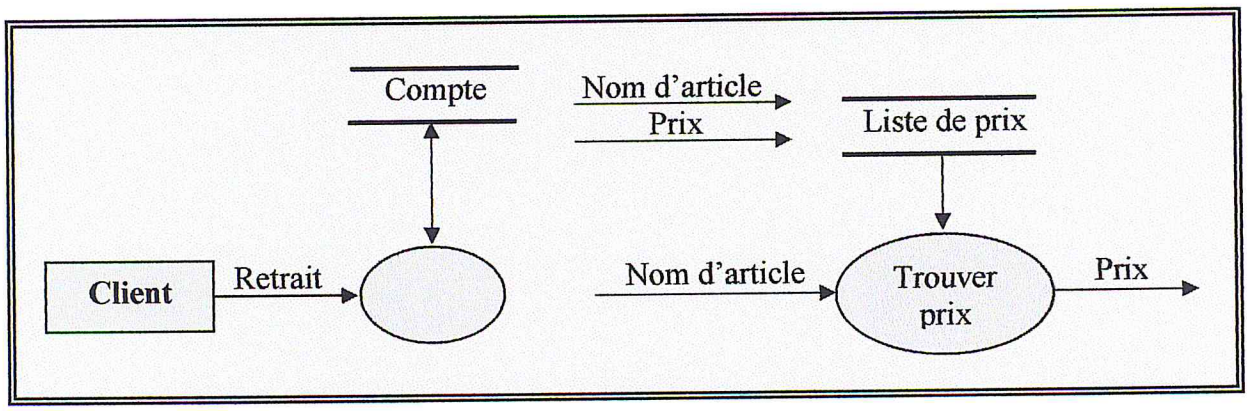


Figure IV.2.21 : *Réservoir de données.*

3.3.2. Les flots de contrôle

Un flot de contrôle est une valeur booléenne qui permet de savoir si un traitement est évalué. Il est représenté par une ligne pointillée partant d'un traitement produisant une valeur booléenne vers le traitement contrôlé.

La figure *IV.2.22* montre un digramme à flots de contrôle pour un emprunt d'ouvrage d'une bibliothèque, contenant toutes les définitions précédentes.

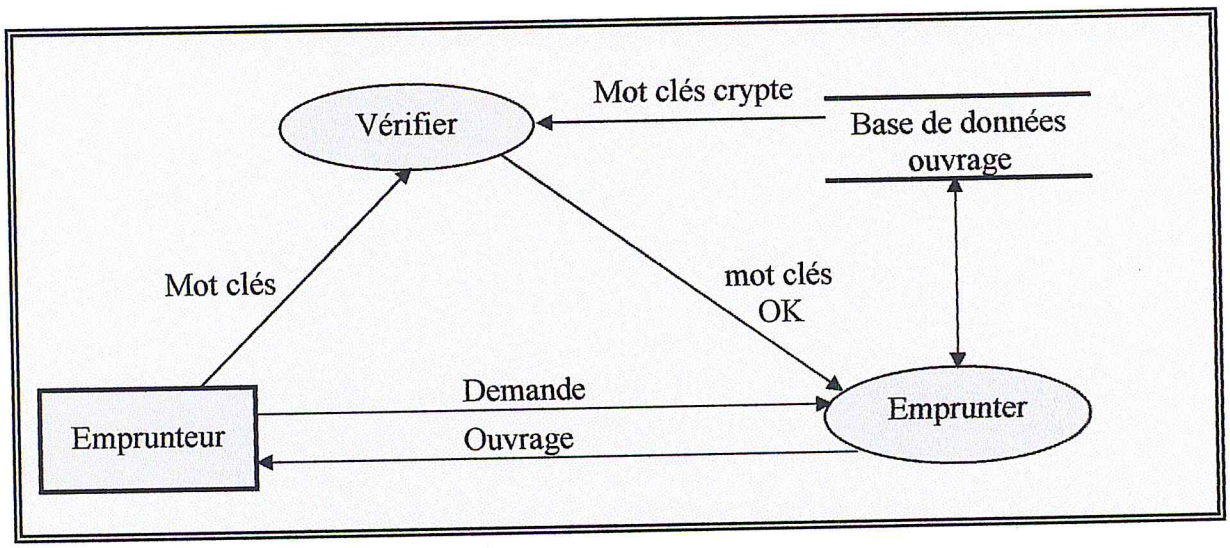


Figure IV.2.22: *Flot de contrôle.*

La démarche méthodologique [BOU 94]

Le cycle développement d'OMT est un cycle en cascade dans lequel les notions de système et de composant sont introduites comme dans le cycle en V

(Figure *IV.2.23*). Mais les seules itérations permises sont vers l'étape antérieure.

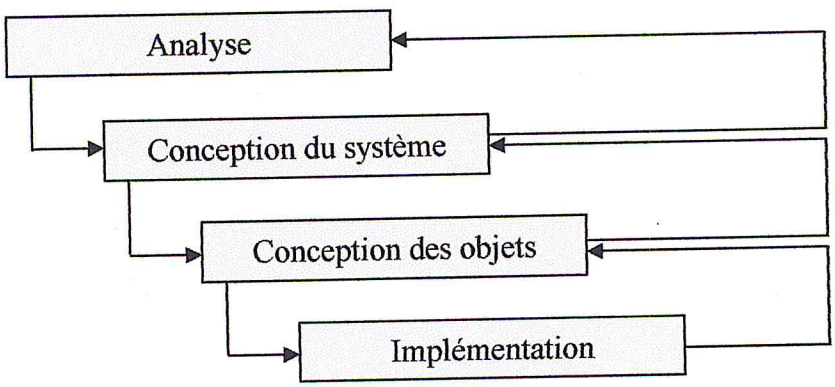


Figure IV.2.23 : *Cycle de développement d'OMT.*

1. L'étape d'analyse

L'étape d'analyse permet d'élaborer les trois modèles conceptuels (statique, dynamique et fonctionnel). Ce niveau est une description de ce que le système doit faire et non du " comment " le faire.

2. L'étape de conception du système

L'étape de conception du système permet de définir l'architecture, les constituants et les ressources du système informatique. Elle permet de décomposer le système d'information en sous-systèmes, d'identifier les éléments de conflits, de choisir la stratégie de stockage d'accès aux données et d'implémenter le contrôle avec ses contraintes et ses exceptions.

3. L'étape de conception des objets

L'étape de conception des objets est une spécification détaillée de l'implémentation des objets, indépendante d'un environnement donné, mais compatible avec une technologie choisie, c'est le niveau logique. C'est à ce niveau que l'ensemble des opérations sur les objets est complètement identifié et spécifié.

Conclusion

On a présenté une méthode de développement d'un logiciel par objets. La méthode OMT est une méthode pareille à toutes les autres méthodes, elle respecte les trois étapes de l'étude : l'étape statique, dynamique et fonctionnelle. Statique pour aboutir à un modèle objet, dynamique pour décrire la réalisation des objets du système aux événements et les itérations entre les objets eux-mêmes, et enfin, l'étape fonctionnelle qui comprend des contraintes entre les valeurs, les informations de contrôle et de structure d'objet appartenant respectivement au modèle dynamique et au modèle objet.

Chapitre V :

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 lesquelles 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.

Partie 1 :

Analyse

I. L'ANALYSE

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

1. Modèle objet

Le modèle objet fournit la structure statique 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 associations (incluant les agrégations) entre objets.
- Identification des attributs de chaque objet.
- Organisation et simplification des classes d'objets en utilisant l'héritage.

1.1. Identification des classes d'objets

La première étape dans la construction d'un modèle objet consiste à identifier les classes d'objets à partir du domaine d'application. La figure ci-dessous illustre les classes utiles de l'étude.

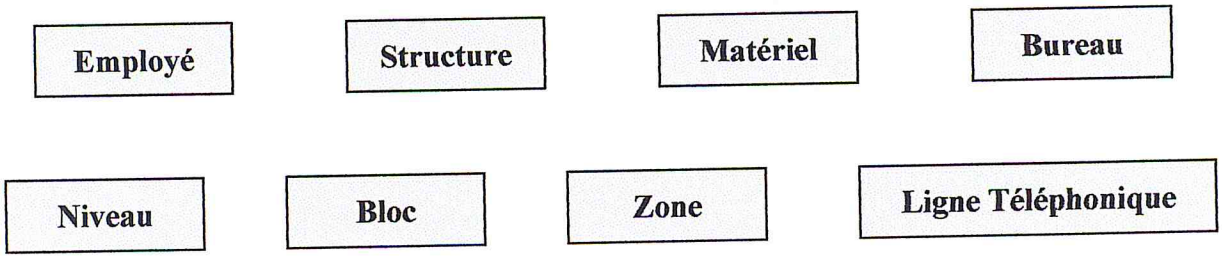


Figure : *les classes*

1.2. Dictionnaire des données

Un dictionnaire de données 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.

Employé : Classe qui identifie les employés.
Structure : Classe qui englobe les différents services de la Sonatrach.
Matériel : Classe qui identifie le matériel de la Sonatrach.
Bureau : Classe qui identifie les différents bureaux appartenant à un service.
Niveau : Classe qui identifie les différents étages de la Sonatrach.
Bloc : Classe qui permet de représenter les blocs de la Sonatrach.
Zone : Classe qui identifie les différentes zones d'un bloc.
Ligne téléphonique : Classe qui identifie le numéro de la ligne téléphonique.

1.3. 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 définies au sein d'une organisation.

Association	Dimension	Cardinalité	Objets intervenant	Propriété de l'association
Occupe	2	Zéro à plusieurs	Structure Bloc	
Possède	2	Zéro à un	Employé Ligne téléphonique	
Contient	2	Zéro à plusieurs	Bureau Matériel	
Est destiné	2	Zéro à plusieurs	Matériel Employé	
Est affecté	2	Zéro à plusieurs	Employé Structure	
Est divisé	2	Un à plusieurs	Bloc Niveau	
Est composé	2	Un à plusieurs	Bloc Zone	
Est composée	2	Un à plusieurs	Zone Bureau	

1.4. Identification des attributs :

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

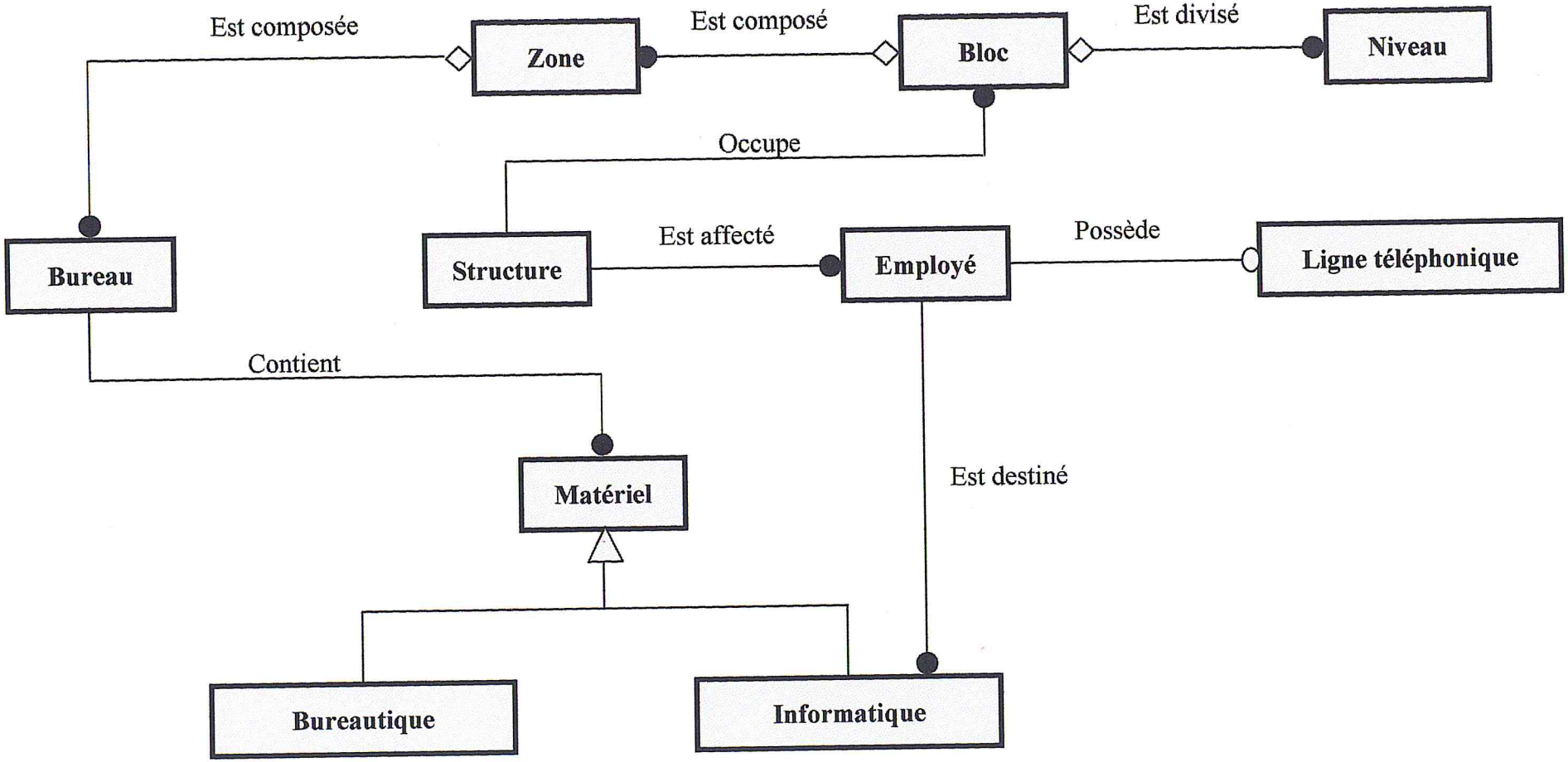
Classe	Attributs	Signification	type	taille
Employé	Mat-empl	Matricule employé	N	08
	Nom-empl	Nom employé	A	15
	Prénom-empl	Prénom employé	A	20
	Nom-j-fille	Nom de jeune fille	A	15
	Date-nais-empl	Date de naissance employé	Date	08
	Lieu-nais-empl	Lieu de naissance employé	A	20
	Sexe	Sexe	A	09
	NAT	Nationalité	A	15
	ADR-empl	Adresse employé	AN	30
	NS	Numéro de sécurité sociale	N	10
	Fct	Fonction	A	35
	SF	Situation familiale	A	1
	Nbr-enf	Nombre d'enfant	N	02
Structure	Code-str	Code structure	A	03
	Design-str	Désignation structure	A	35
	Miss_str	Mission structure	A	50
	Nbre-empl	Nombre d'employés	N	03
Matériel	Code-mat	Code matériel	N	20
	Design-mat	Désignation matériel	A	20
	Type-mat	Type matériel	A	12
	Config	configuration	AN	50
Bureau	Num-bur	Numéro bureau	AN	08
	Surface	Surface	N	02
	Lign-reseau	Ligne réseau	AN	08
Niveau	Num-niv	Numéro niveau	N	02
Bloc	Num-bloc	Numéro bloc	A	01
	Nbre-zone	Nombre zones	N	01
Zone	Num-zone	Numéro zone	N	01
	Nbre-bur	Nombre bureaux	N	02
Ligne téléphonique	Num-poste	Numéro poste	AN	08

1.5. Utilisation de l'héritage [BOUZ 94]

L'étape suivante consiste à réorganiser les classes à l'aide de l'héritage afin de partager les structures communes.

Matériel : classe représentant les propriétés communes de Matériel informatique et Matériel bureautique.

Modèle Objet



2. Modèle Dynamique

- Développement d'un modèle dynamique

Le modèle dynamique décrit des concepts traitant du flot de contrôle des interactions et des séquences d'opérations dans un système d'objet actif (concurrents). **[RUM 95]**

Nous rappelons les deux concepts suivants :

Evènement : un évènement 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'évènements se déroulant durant une exécution particulière d'un système. Chaque événement transmet une information d'un objet à un autre.

Préparation des scénarios

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 :

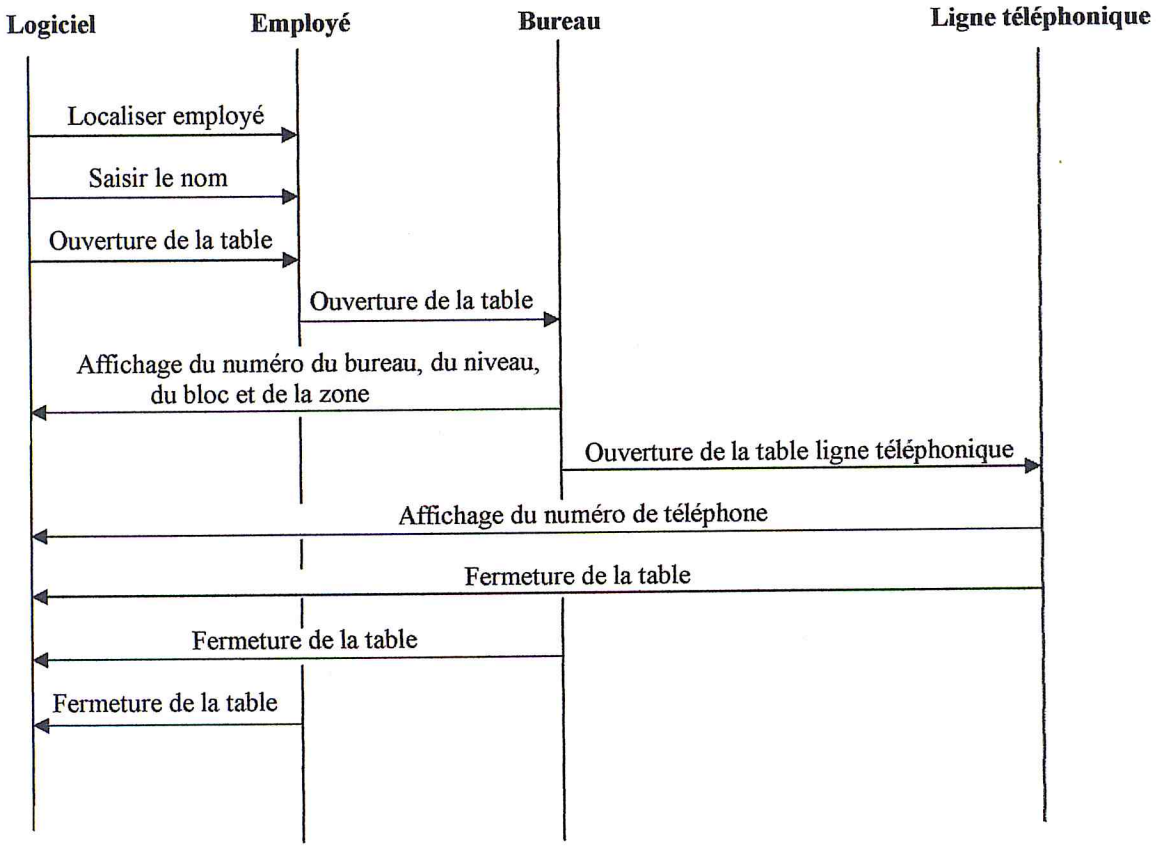
- Localisation d'un employé.
- Changement d'une structure.
- Changement de fonction.
- Quitter un poste de travail.
- Localisation d'un matériel (Informatique et Bureautique).
- Changement de bureau pour un matériel.
- Ajout d'une nouvelle ligne téléphonique.
- Changement d'une ligne téléphonique.
- Suppression d'une ligne téléphonique.
- Ajout d'une nouvelle structure.

2.1. Scénarios et suivi d'événements

Scénario Localisation d'un employé (cas normal).

- Choisir, localiser employé.
- Une page de recherche est ouverte.
- Saisir le nom de l'employé.
- Lancer la recherche.
- La table bureau est ouverte.
- Le bureau de l'employé clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Une page est ouverte avec le numéro de la ligne téléphonique.
- Cliquer sur quitter pour fermer la table ligne téléphonique et la table bureau.
- Affichage d'un message qui demande si il y a une autre recherche à faire.
- Cliquer sur non pour quitter.

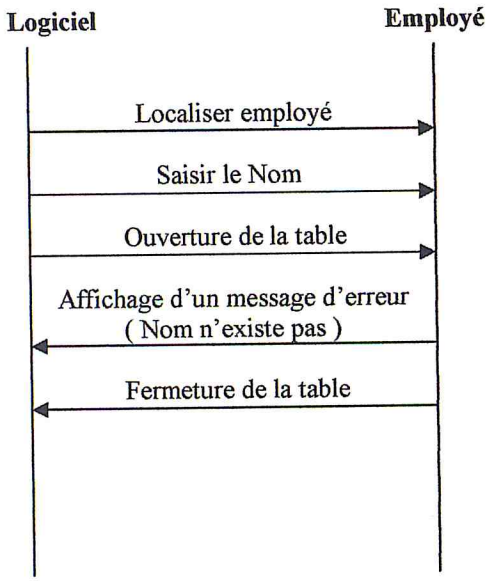
Suivi d'événement :



Scénario Localisation d'un employé (cas d'erreur).

- Choisir, localiser employé.
- La page de recherche est ouverte.
- Saisir le nom de l'employé.
- Lancer la recherche.
- L'employé n'existe pas.
- Affichage d'un message d'erreur.
- Cliquer sur fermer pour quitter.

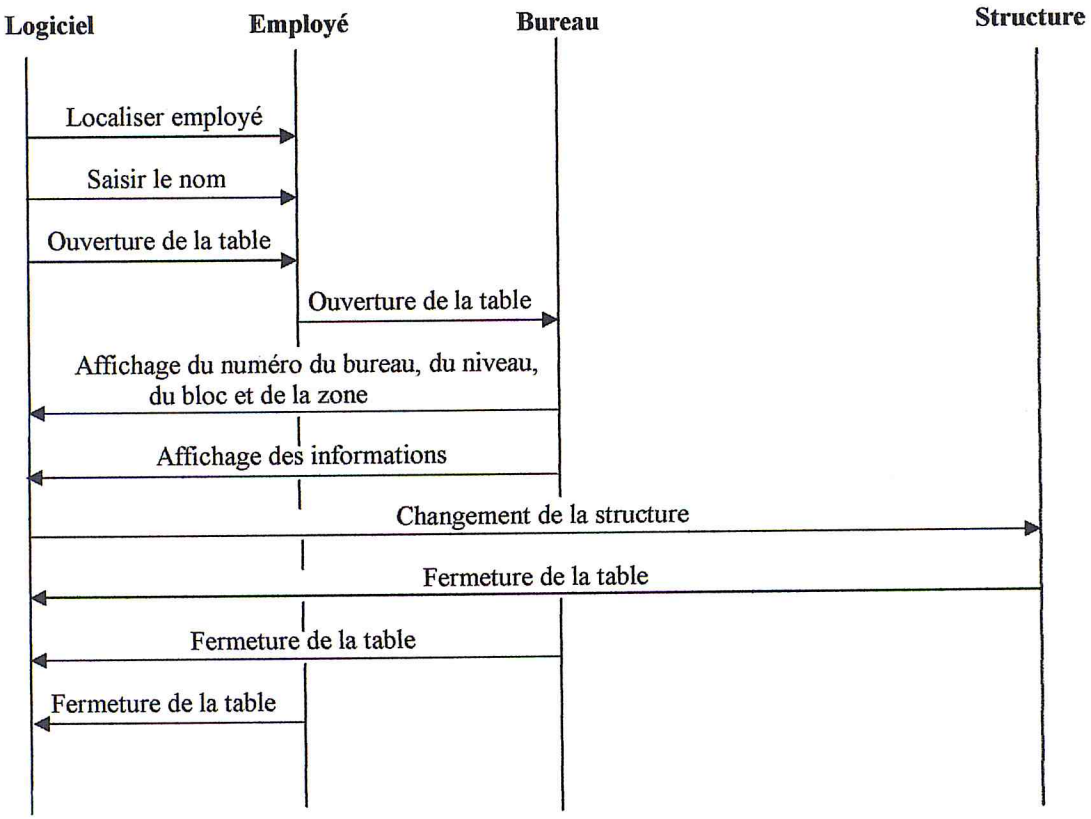
Suivi d'événement :



Scénario Changement de structure pour un employé :

- Choisir, localiser employé.
- La page de recherche est ouverte.
- Saisir le nom de l'employé.
- Lancer la recherche.
- La table bureau est ouverte.
- Le bureau de l'employé clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Cliquer sur le matricule de l'employé.
- Une page s'affiche avec toutes les informations attribuées à l'employé.
- Changer la structure affectée.
- Cliquer sur modification.
- Affichage d'un message de confirmation.
- Cliquer sur Oui.
- Affichage d'un message qui demande s'il y a une autre modification à faire.
- Cliquer sur Non pour quitter.

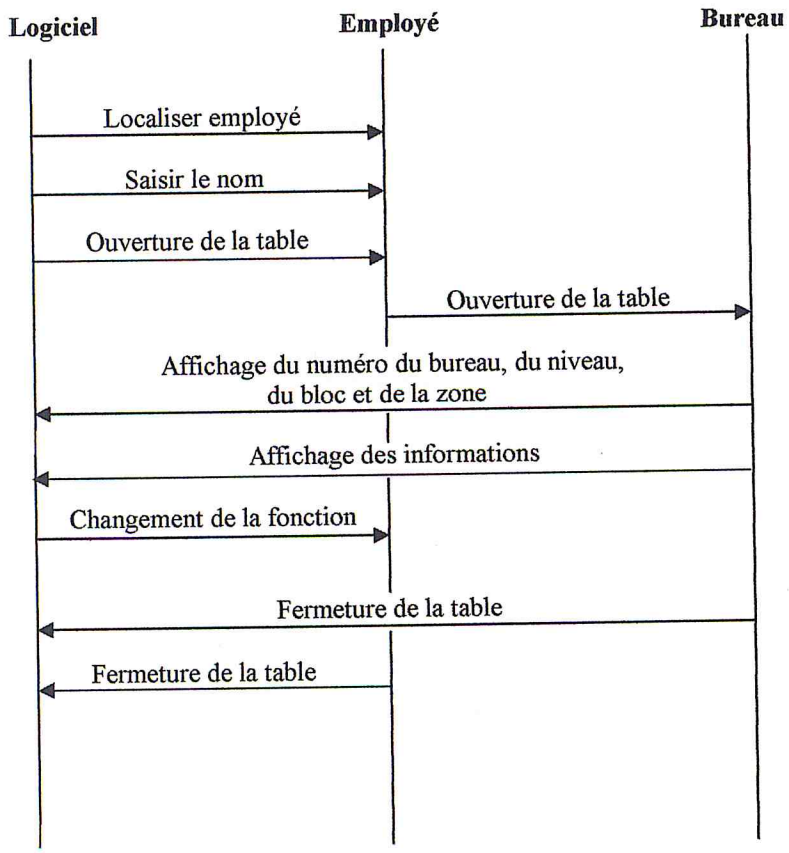
Suivi d'événement :



Scénario Changement de fonction pour un employé :

- Choisir, localiser employé.
- La page de recherche est ouverte.
- Saisir le nom de l'employé.
- Lancer la recherche.
- La table bureau est ouverte.
- Le bureau de l'employé clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Cliquer sur le matricule de l'employé.
- Une page s'affiche avec toutes les informations attribuées à l'employé.
- Changer la fonction.
- Cliquer sur modification.
- Affichage d'un message de confirmation.
- Cliquer sur Oui.
- Affichage d'un message qui demande s'il y a une autre modification à faire.
- Cliquer sur Non pour quitter.

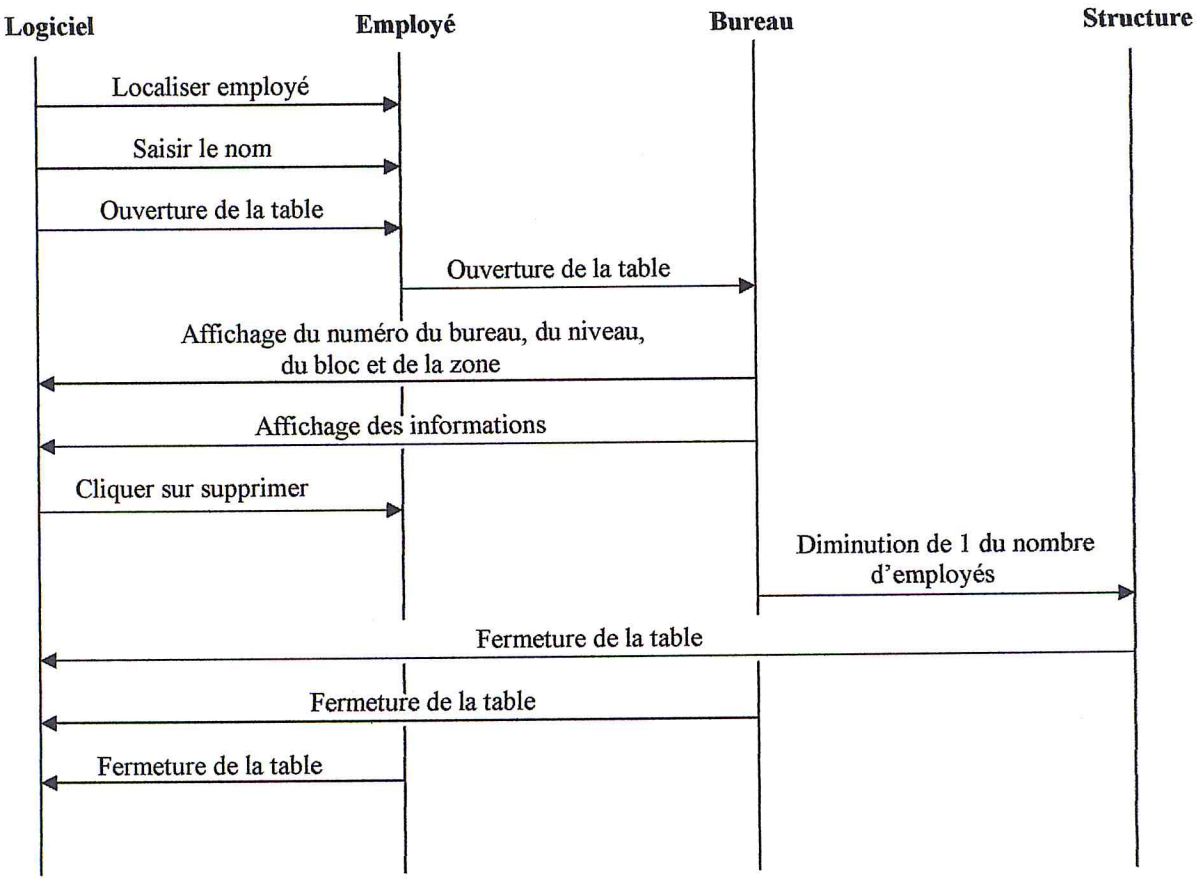
Suivi d'événement :



Scénario Quitter un poste de travail :

- Choisir, localiser employé.
- Une page de recherche est ouverte.
- Saisir le nom de l'employé.
- Lancer la recherche.
- La table bureau est ouverte.
- Le bureau de l'employé clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Cliquer sur le matricule de l'employé.
- Une page s'affiche avec toutes les informations attribuées à l'employé.
- Cliquer sur supprimer.
- Affichage d'un message d'avertissement.
- Cliquer sur Oui pour confirmer.
- Affichage d'un message de confirmation.
- Cliquer sur Oui pour confirmer.
- Automatiquement la table Structure est modifiée (le nombre d'employé diminue de 1).
- Affichage d'un message qui demande s'il y a une autre suppression à faire.
- Cliquer sur Non pour quitter.
- Fermeture de la table employé.

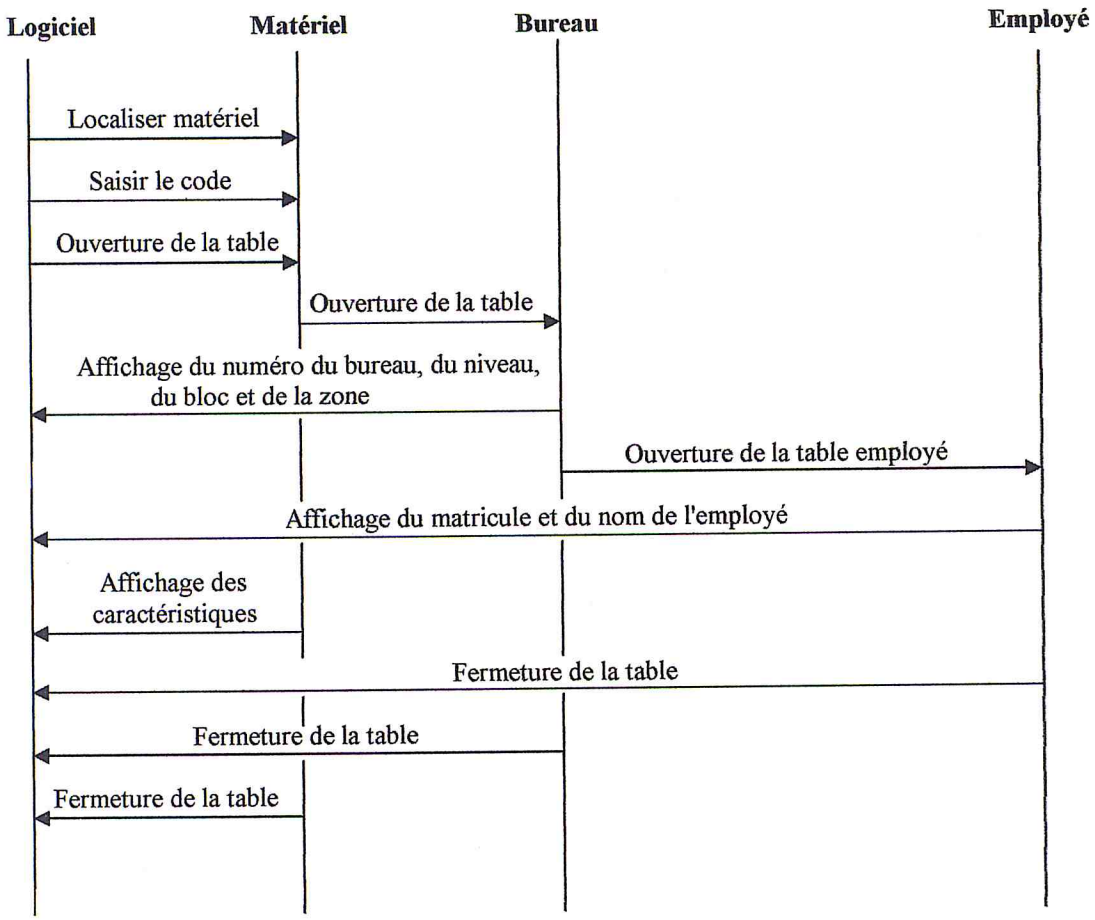
Suivi d'événement :



Scénario Localisation d'un matériel Informatique

- Choisir, localiser Matériel.
- Une page de recherche est ouverte.
- Saisir le code du matériel.
- Lancer la recherche.
- Le bureau correspondant clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Affichage des caractéristiques du matériel, le matricule et le nom de la personne a qui le matériel est destiné.
- Cliquer sur Fermer pour quitter.
- Affichage d'un message qui demande s'il y a un autre matériel a localiser.
- Cliquer sur Non pour quitter.

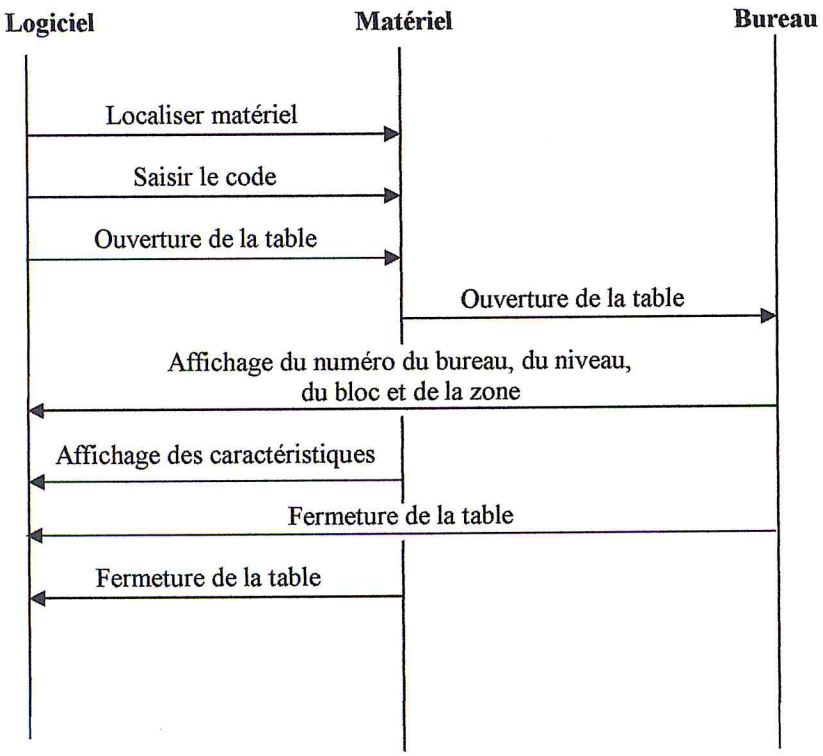
Suivi d'événement :



Scénario Localisation d'un matériel Bureautique

- Choisir, localiser Matériel.
- Une page de recherche est ouverte.
- Saisir le code du matériel.
- Lancer la recherche.
- Le bureau correspondant clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Affichage des caractéristiques du matériel.
- Cliquer sur Fermer pour quitter.
- Affichage d'un message qui demande s'il y a un autre matériel a localiser.
- Cliquer sur Non pour quitter.

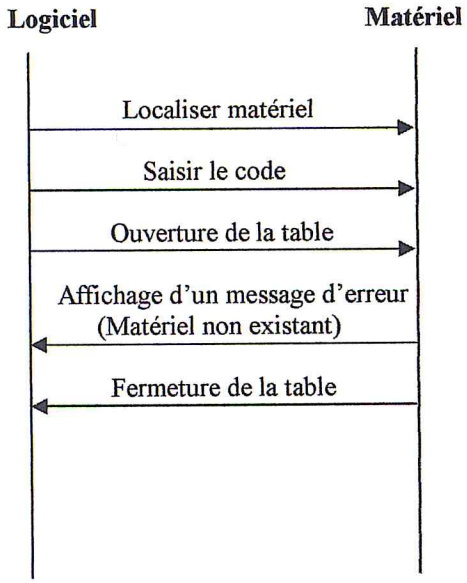
Suivi d'événement :



Scénario Localisation d'un matériel (Cas d'erreur)

- Choisir, localiser Matériel.
- Une page de recherche est ouverte.
- Saisir le code du matériel.
- Lancer la recherche.
- Le matériel n'existe pas.
- Affichage d'un message d'erreur.
- Cliquer sur Fermer pour quitter.

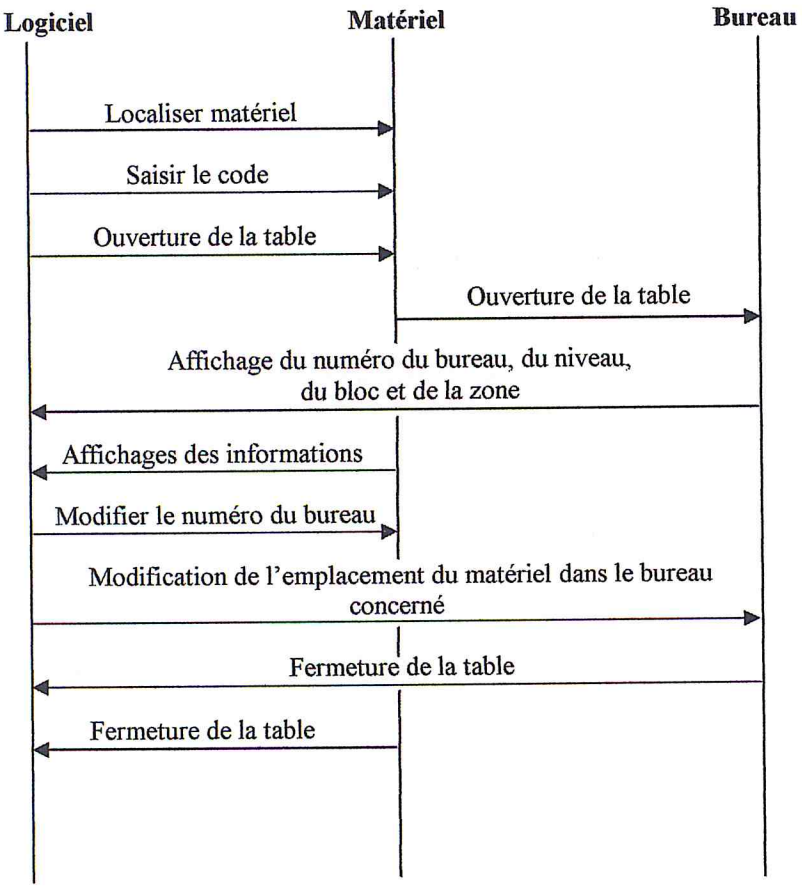
Suivi d'événement



Scénario Changement de bureau pour un matériel

- Choisir, modification matériel.
- Une page de recherche est ouverte.
- Saisir le code du matériel.
- Lancer la recherche.
- Le bureau correspondant clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Affichage des caractéristiques et des différentes informations concernant le matériel.
- Saisir le numéro du nouveau bureau auquel le matériel est affecté.
- Cliquer sur OK pour enregistrer.
- Affichage d'un message d'avertissement qui demande la confirmation.
- Cliquer sur Oui pour confirmer.
- Automatiquement la table bureau est modifiée (le matériel est effacé de l'ancien bureau et enregistré dans le nouveau).
- Affichage d'un message qui demande si on veut faire une autre modification.
- Cliquer sur Non pour quitter.

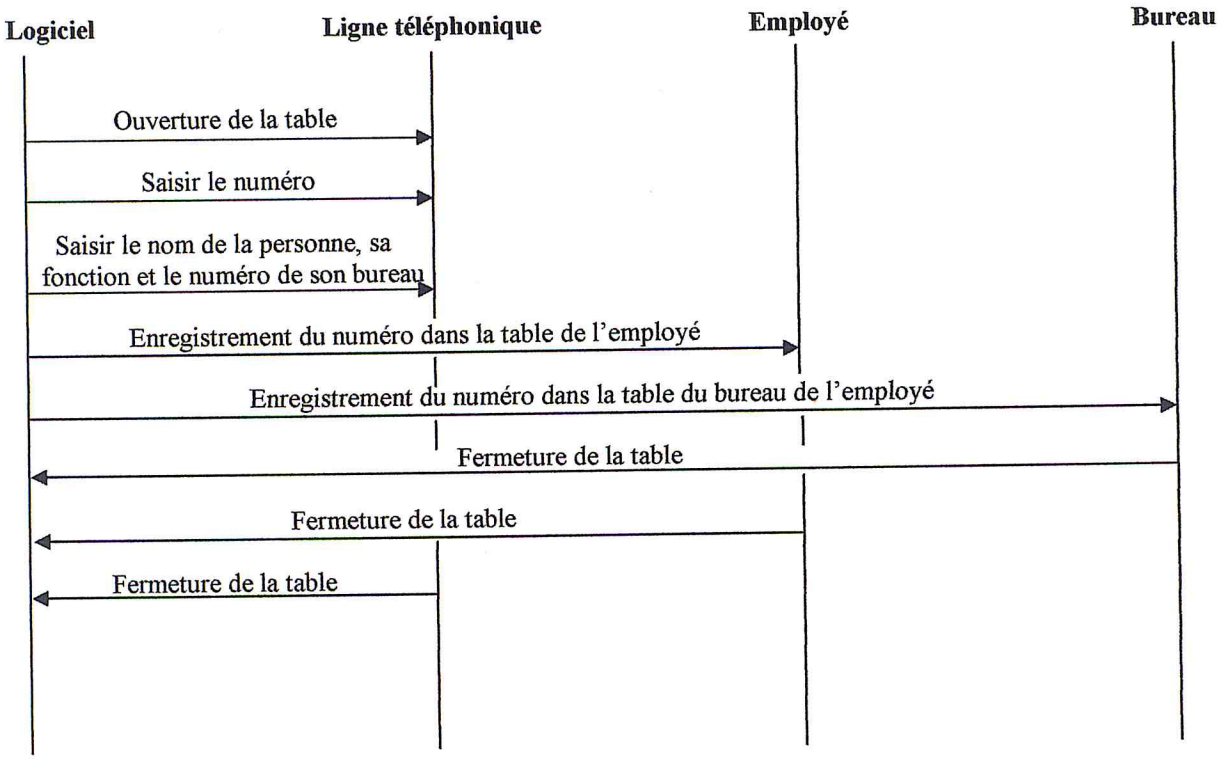
Suivi d'événement :



Scénario Ajout d'une nouvelle ligne téléphonique

- Ouvrir la table Ligne téléphonique.
- Cliquer sur nouvelle ligne.
- Affichage d'une table.
- Saisir le numéro.
- Saisir le nom de la personne a qui le numéro va être attribuer, sa fonction et son numéro de bureau.
- Cliquer sur OK.
- Affichage d'un message de confirmation.
- Cliquer sur Oui pour confirmer.
- Automatiquement la table Employé est modifiée (le nouveau numéro est enregistré dans la table de la personne a qui il est attribué).
- Modification de la table Bureau (ajout du numéro de téléphone dans le bureau de l'employé concerné).
- Affichage d'un message qui demande s'il y a une autre ligne à rajouter.
- Cliquer sur Non pour quitter.
- Fermeture de la table Ligne téléphonique.

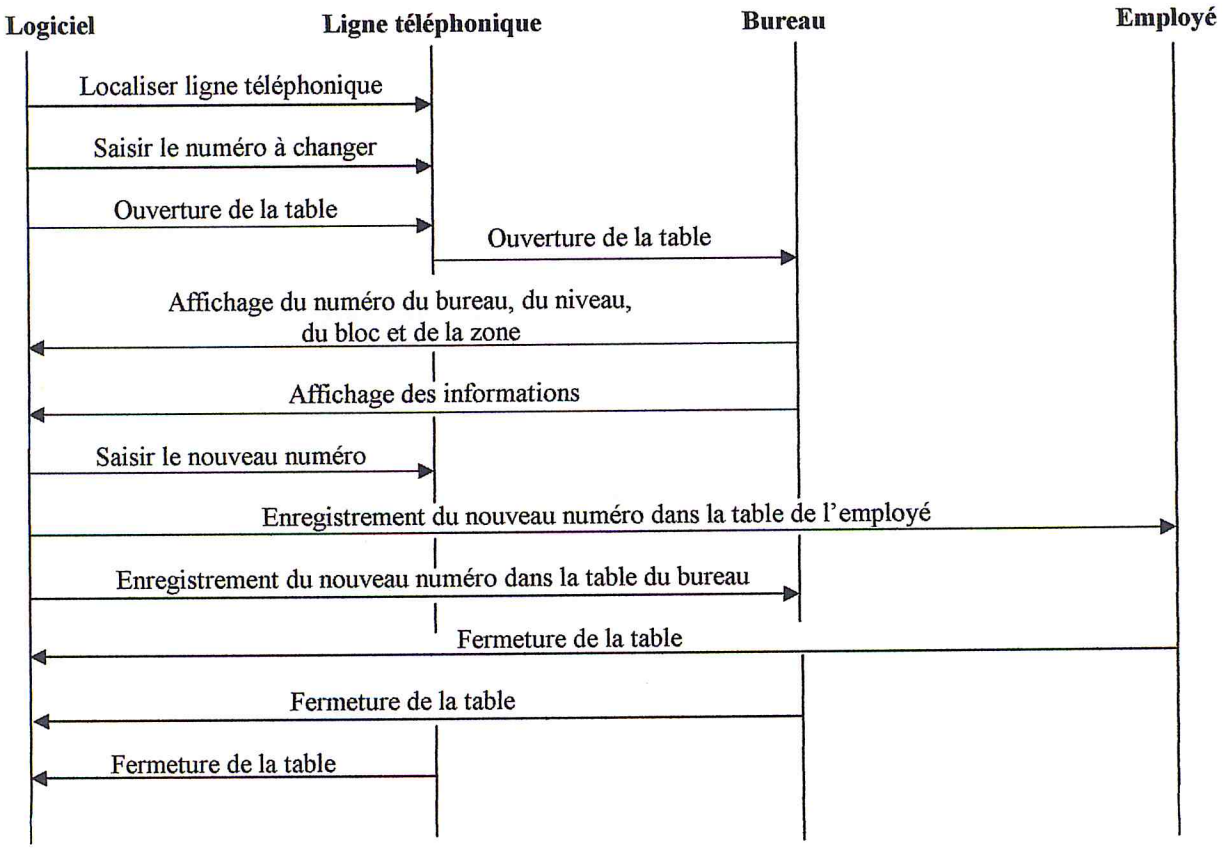
Suivi d'événement



Scénario Changement d'une ligne téléphonique pour un employé

- Choisir, localiser ligne téléphonique.
- Une page de recherche est ouverte.
- Saisir le numéro de la ligne.
- Lancer la recherche.
- La table bureau est ouverte.
- Le bureau correspondant clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Affichage des informations.
- Saisir le nouveau numéro.
- Cliquer sur modifier.
- Affichage d'un message de confirmation.
- Cliquer sur Oui pour confirmer.
- Automatiquement la table Employé est modifiée (le nouveau numéro est enregistré à la place de l'ancien).
- Modification de la table Bureau (le nouveau numéro est enregistré à la place de l'ancien).
- Affichage d'un message qui demande s'il y a un autre changement a faire.
- Cliquer sur Non pour quitter.
- Fermeture de la table Ligne téléphonique.

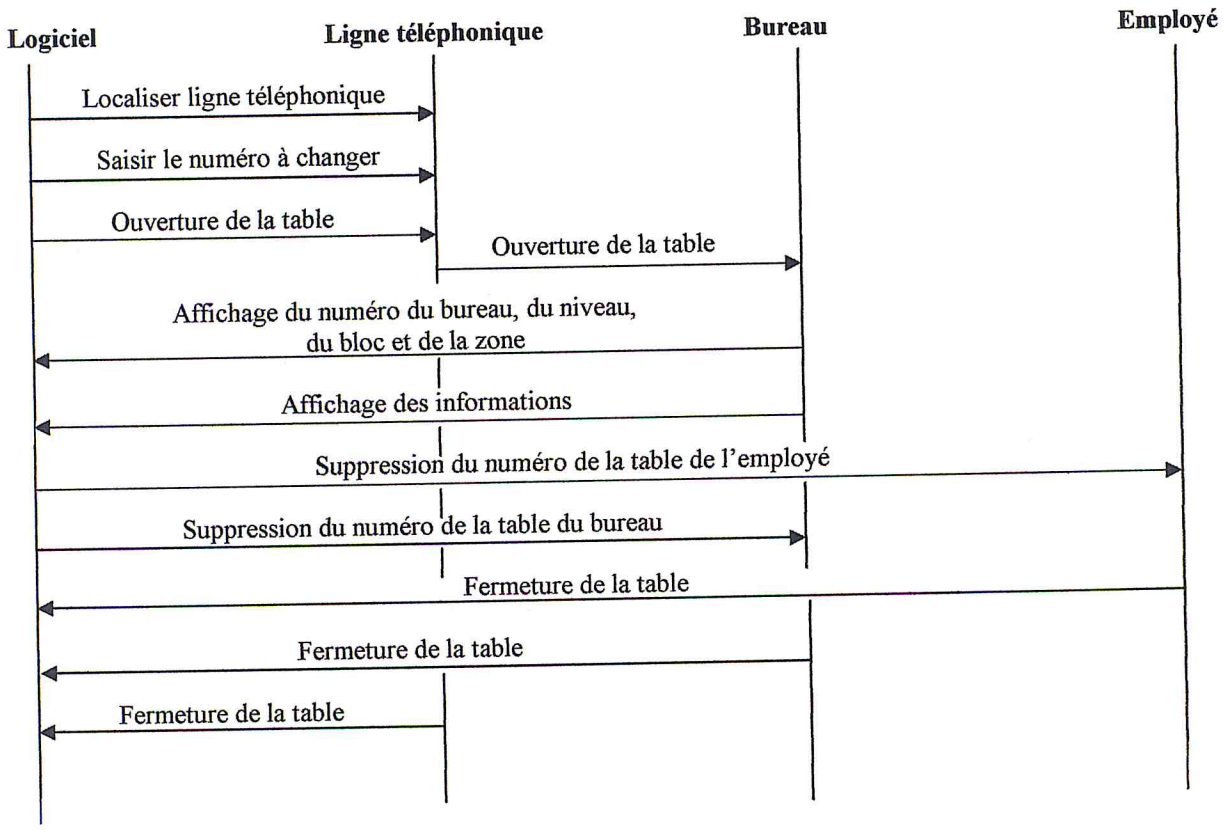
Suivi d'événement



Scénario Suppression d'une ligne téléphonique

- Choisir, localiser ligne téléphonique.
- Une page de recherche est ouverte.
- Saisir le numéro de la ligne.
- Lancer la recherche.
- La table bureau est ouverte.
- Le bureau correspondant clignote.
- Affichage du numéro du niveau, du bloc et de la zone.
- Cliquer sur le bureau.
- Affichage des informations.
- Cliquer sur supprimer.
- Affichage d'un message de confirmation.
- Cliquer sur Oui pour confirmer.
- Automatiquement la table Employé est modifiée (le numéro n'est plus attribué à l'employé).
- Modification de la table Bureau (suppression du numéro de téléphone dans le bureau de l'employé concerné).
- Affichage d'un message qui demande s'il y a une autre ligne à supprimer.
- Cliquer sur Non pour quitter.
- Fermeture de la table Ligne téléphonique.

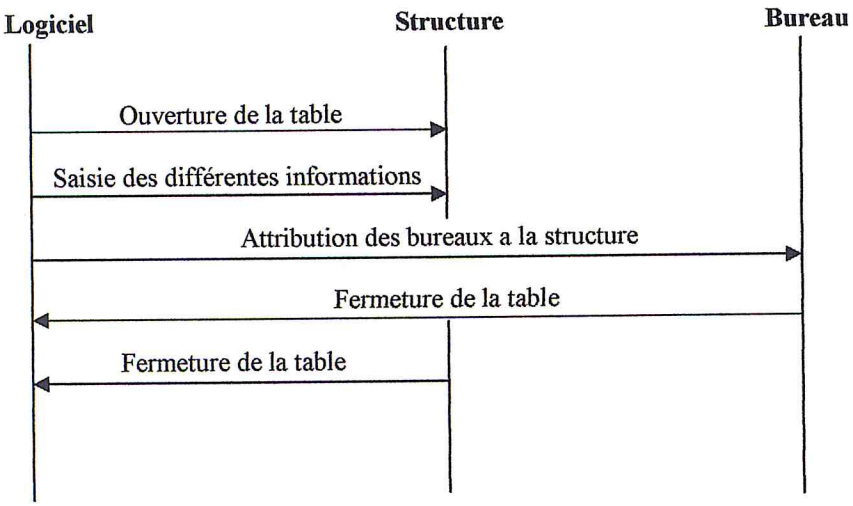
Suivi d'événement



Scénario Ajout d'une nouvelle structure

- Ouvrir la table Structure.
- Cliquer sur Nouvelle.
- Affichage d'une table vide.
- Saisir les informations concernant la nouvelle structure (code, désignation, mission, nombre de personnel, les étages occupés, les blocs et les zones).
- Cliquer sur enregistrer.
- Affichage d'un message de confirmation.
- Cliquer sur Oui pour confirmer.
- Automatiquement la table Bureau est modifiée (attribution des bureaux à la nouvelle structure).
- Affichage d'un message qui demande s'il y a une autre structure à saisir.
- Cliquer sur Non pour quitter.
- Fermeture de la table Bureau.
- Fermeture de la table Structure.

Suivi d'événement



2.2. Diagramme d'état

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

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

Etat

C'est une abstraction des valeurs des attributs et des liens d'un objet, il spécifie la réponse de l'objet aux évènements d'entrées.

Les conditions

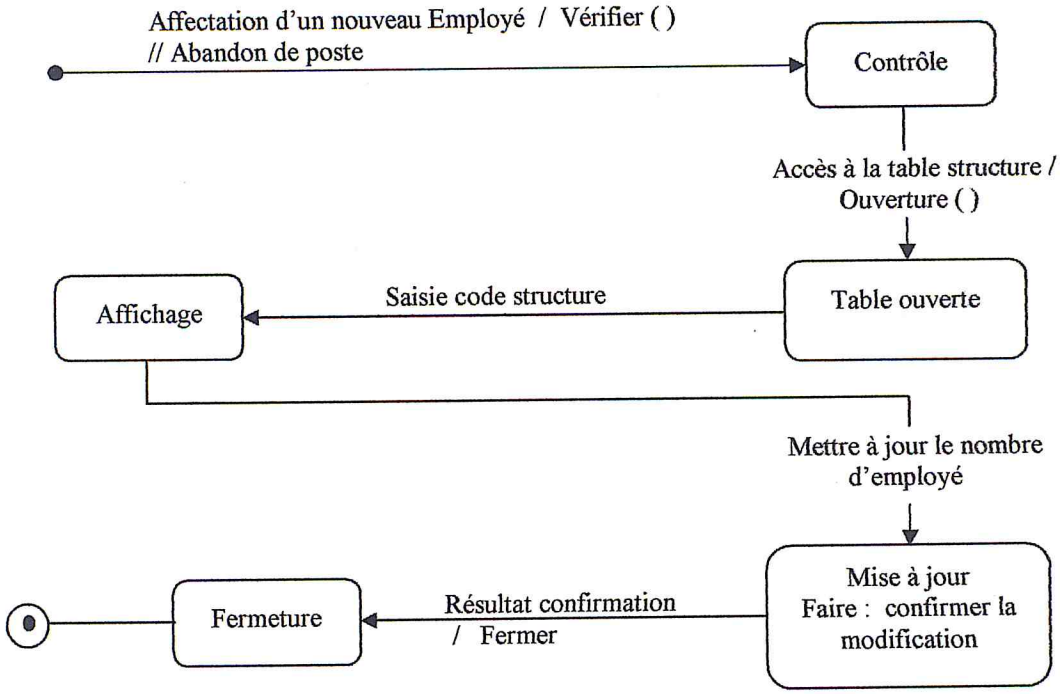
Une condition et une fonction booléenne utilisée comme grade sur des transitions, elle est représentée entre crochets suivant le nom d'évènement.

Les opérations

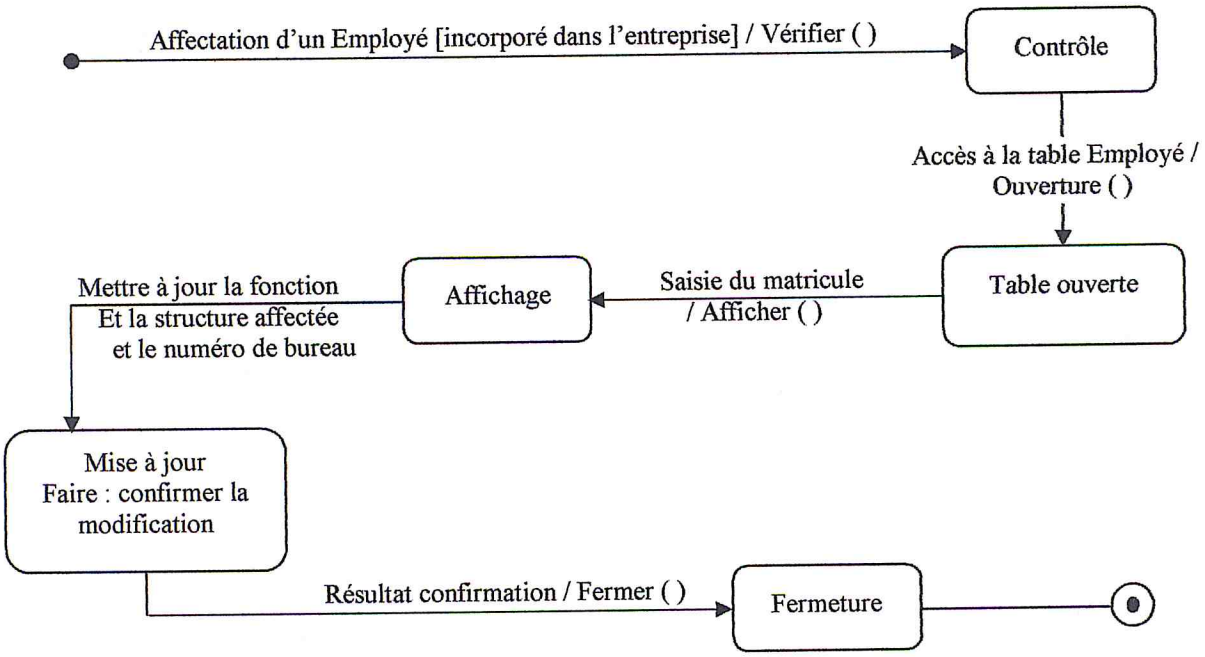
C'est des réponses à des évènements (action) ou à des états (activité). L'activité est représentée par la notation « faire : A », l'action est représentée par une barre optique « / » et le nom de l'action suivant le nom de l'événement.

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

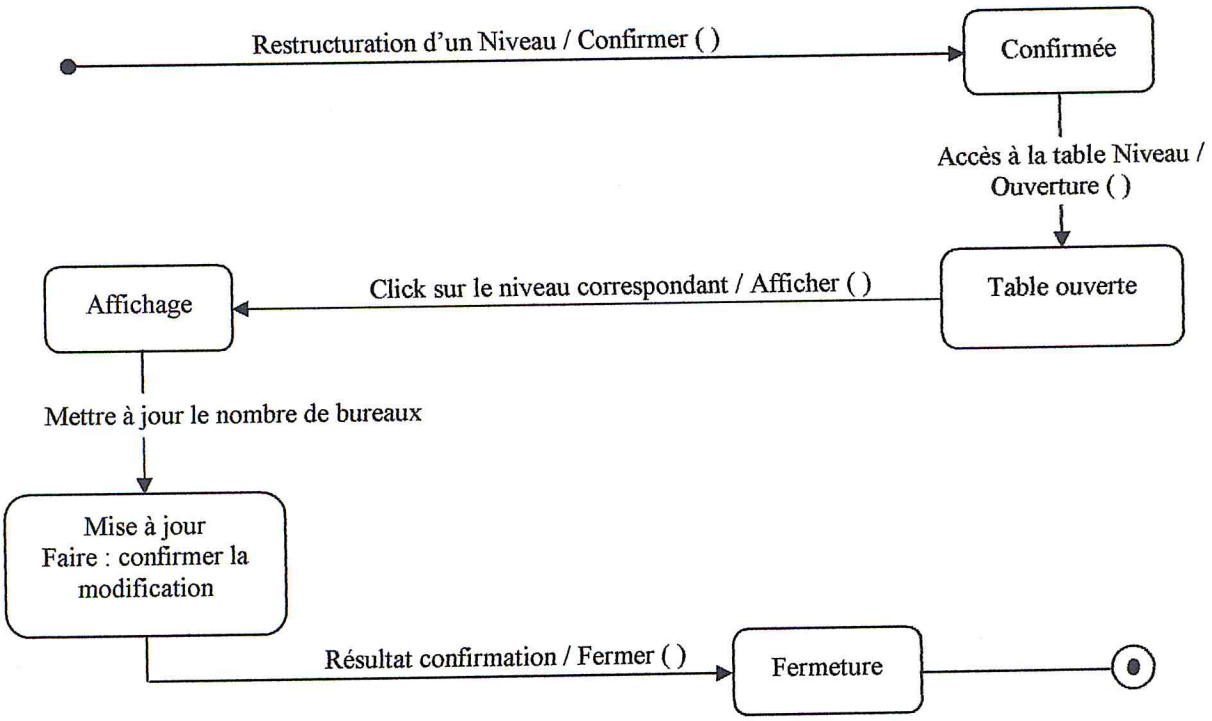
2.2.1. Diagramme d'état d'une Structure



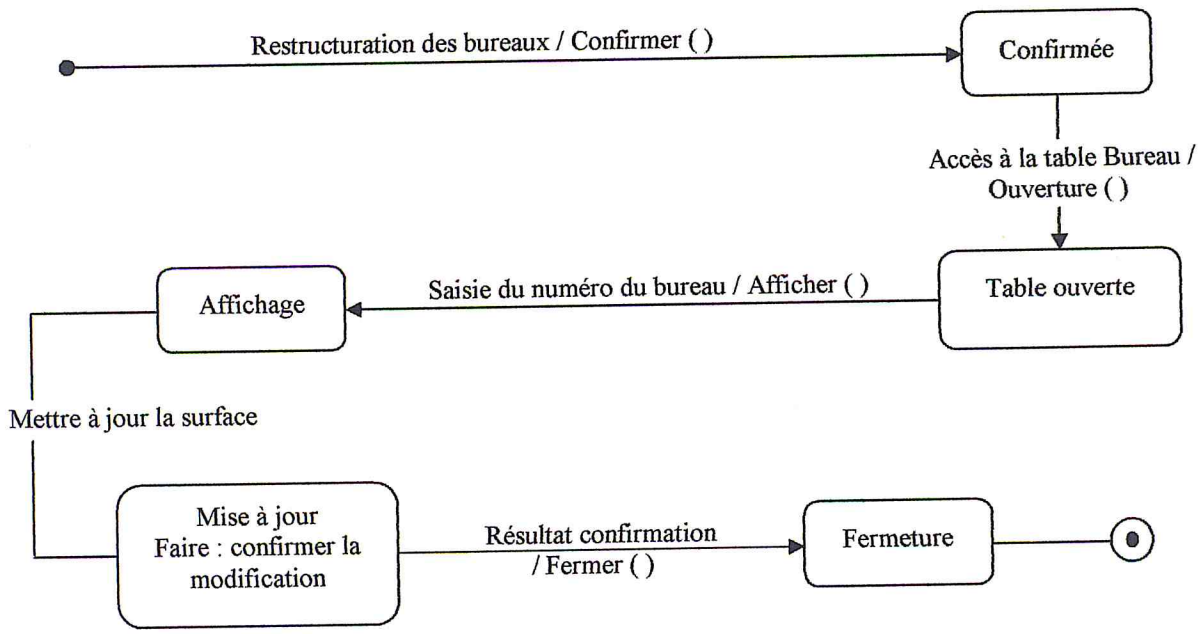
2.2.2. Diagramme d'état d'un Employé



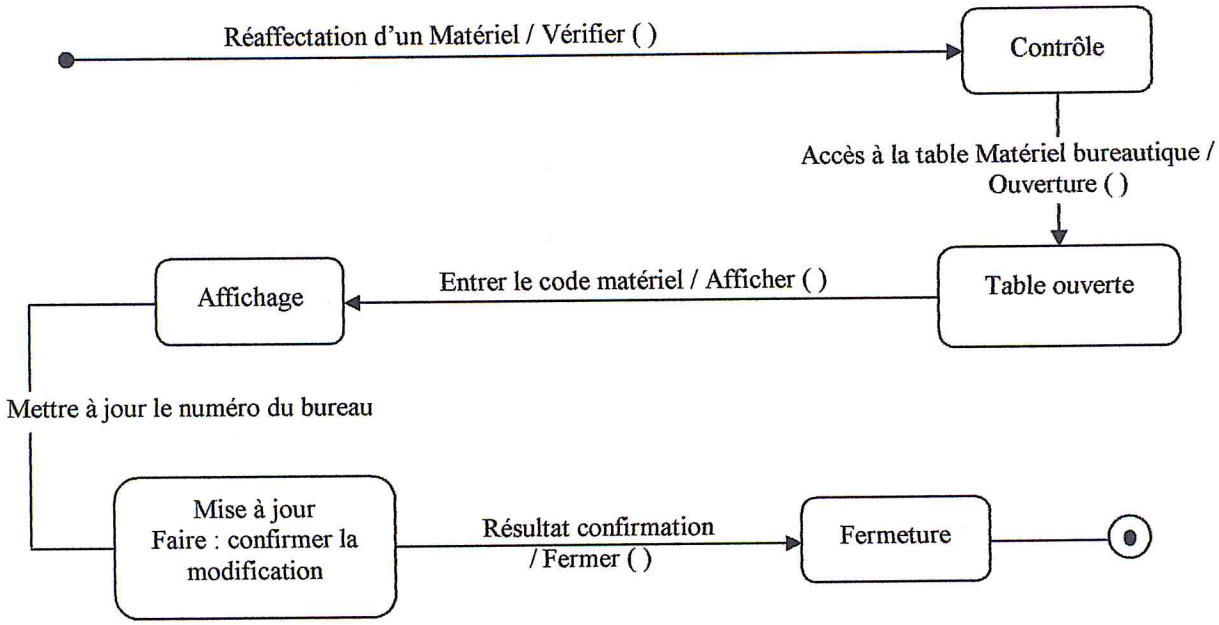
2.2.3. Diagramme d'état d'un Niveau



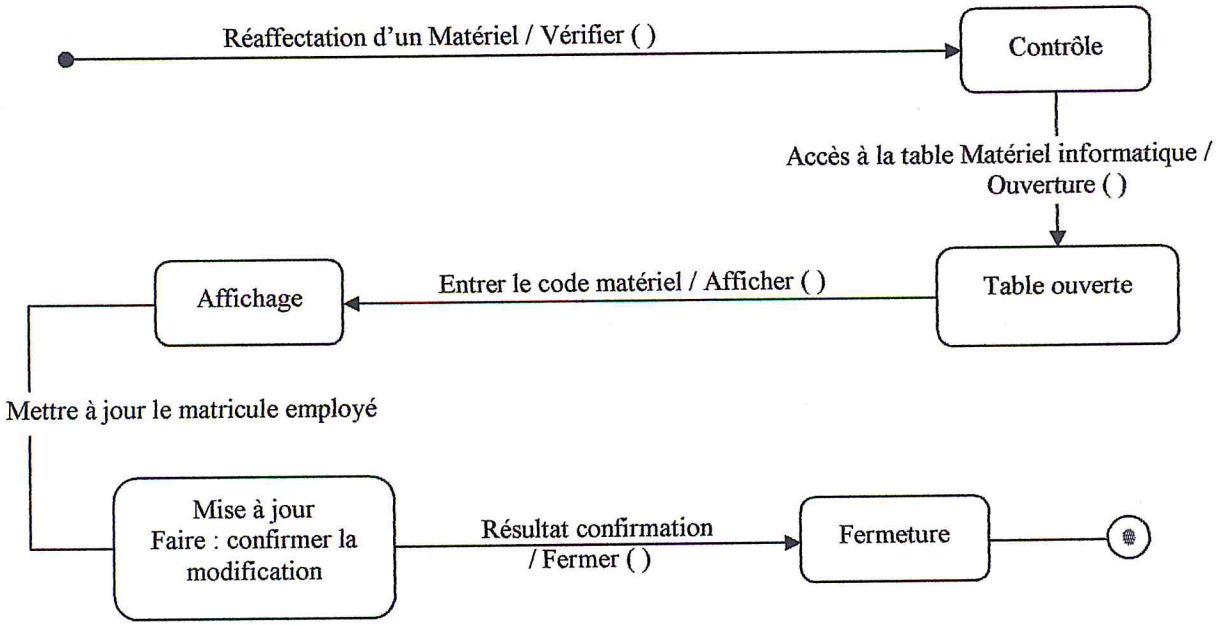
2.2.4. Diagramme d'état d'un Bureau



2.2.5. Diagramme d'état d'un Matériel Bureautique



2.2.5. Diagramme d'état d'un Matériel Informatique



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.

1. 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 : **[RUM 95]**

- Fonction

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

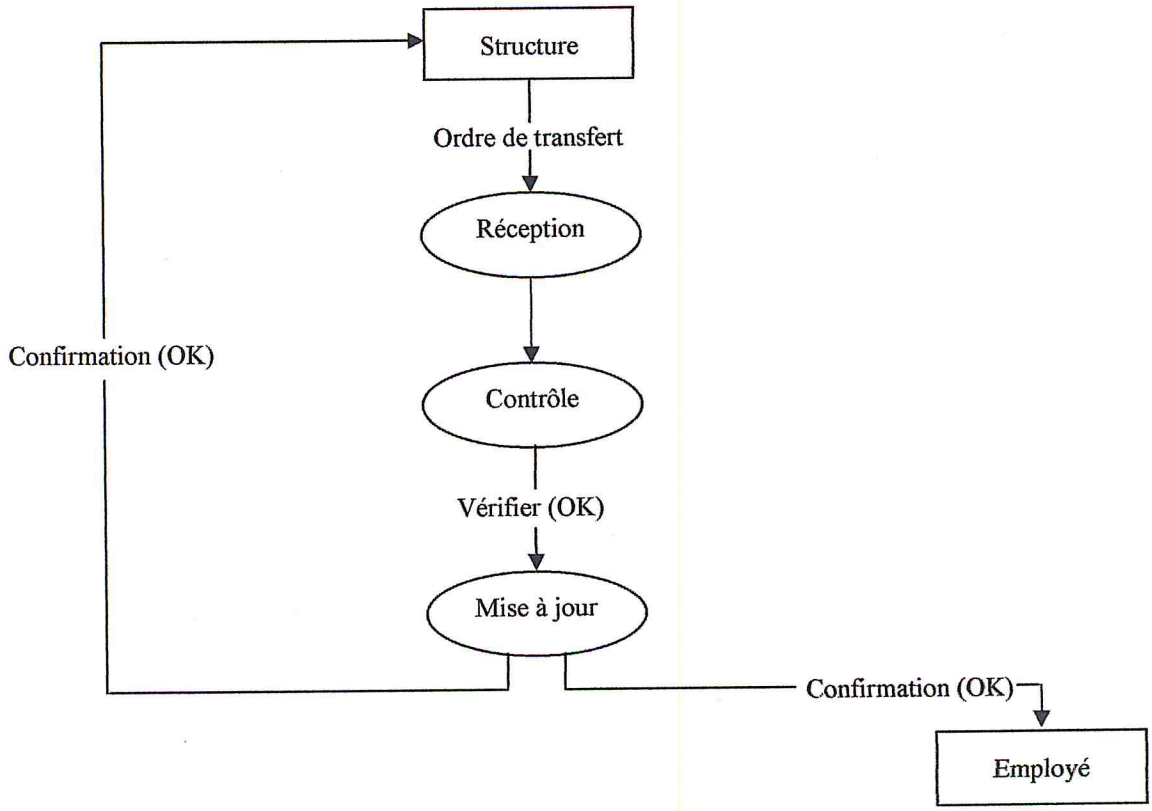
- Contrainte

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.

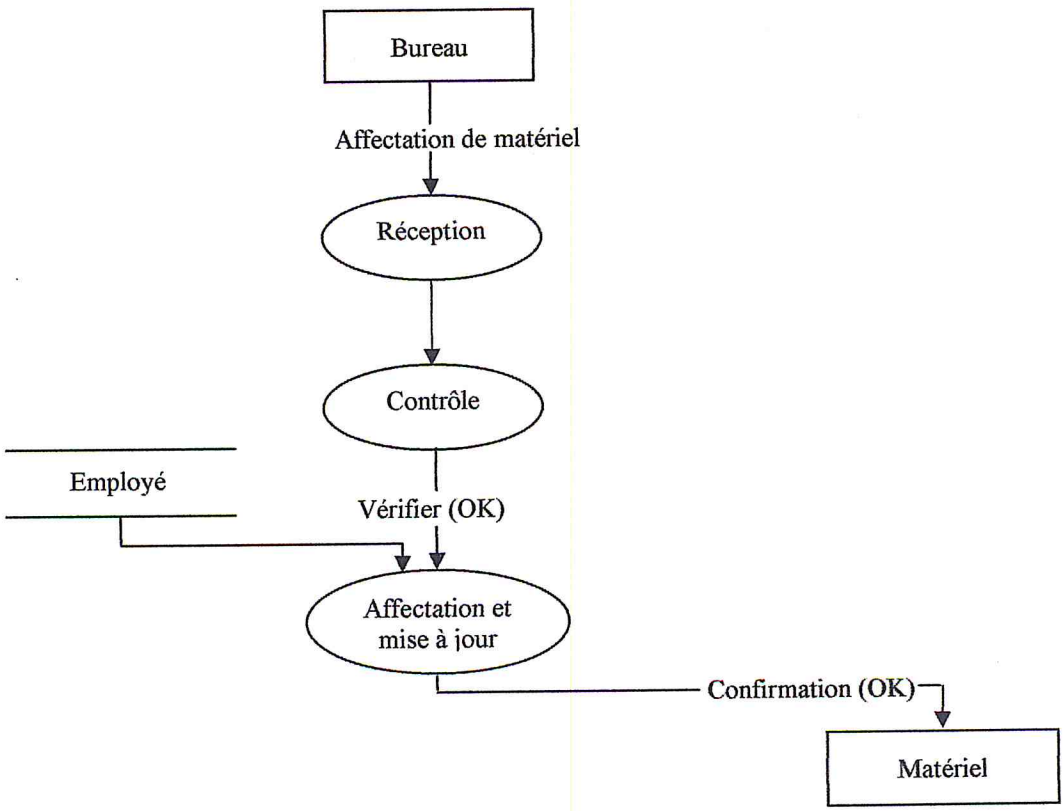
Les DFD correspondants à notre système seront les suivants :

3.1. Diagramme à flots de données

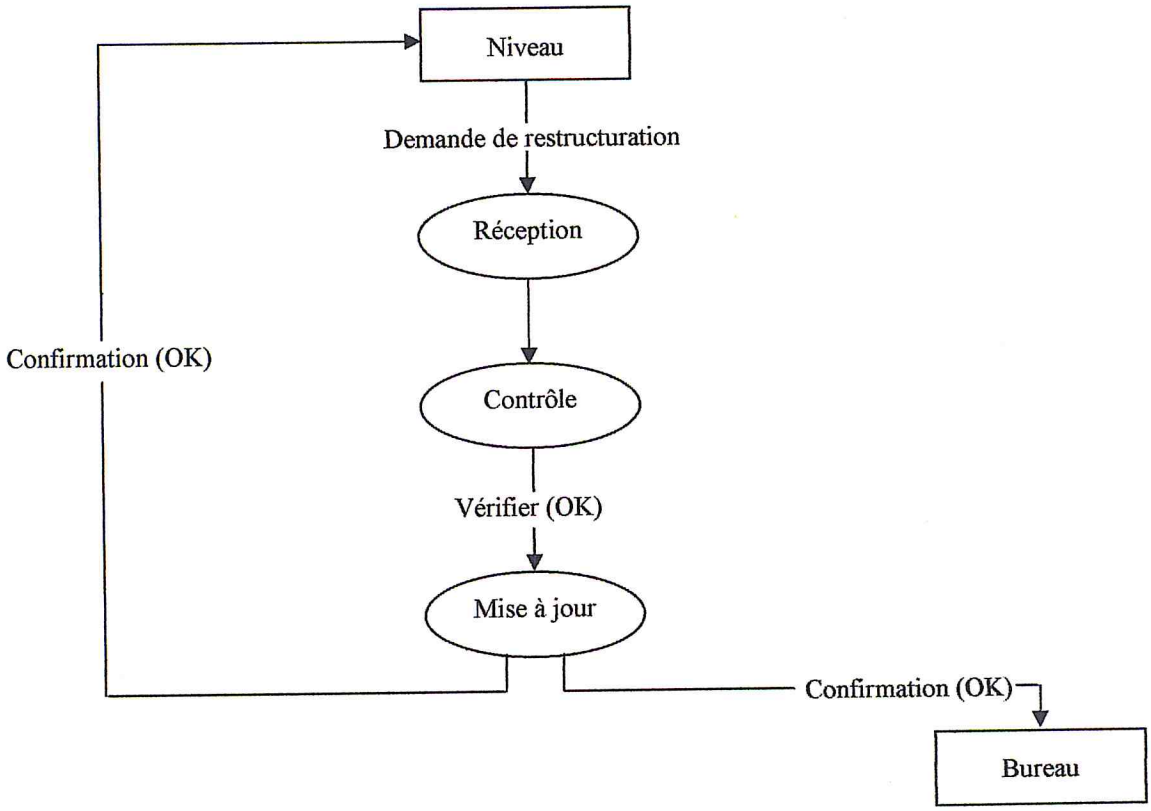
3.1.1. Diagramme à flots de données du traitement « Etablir le transfert d'un employé »



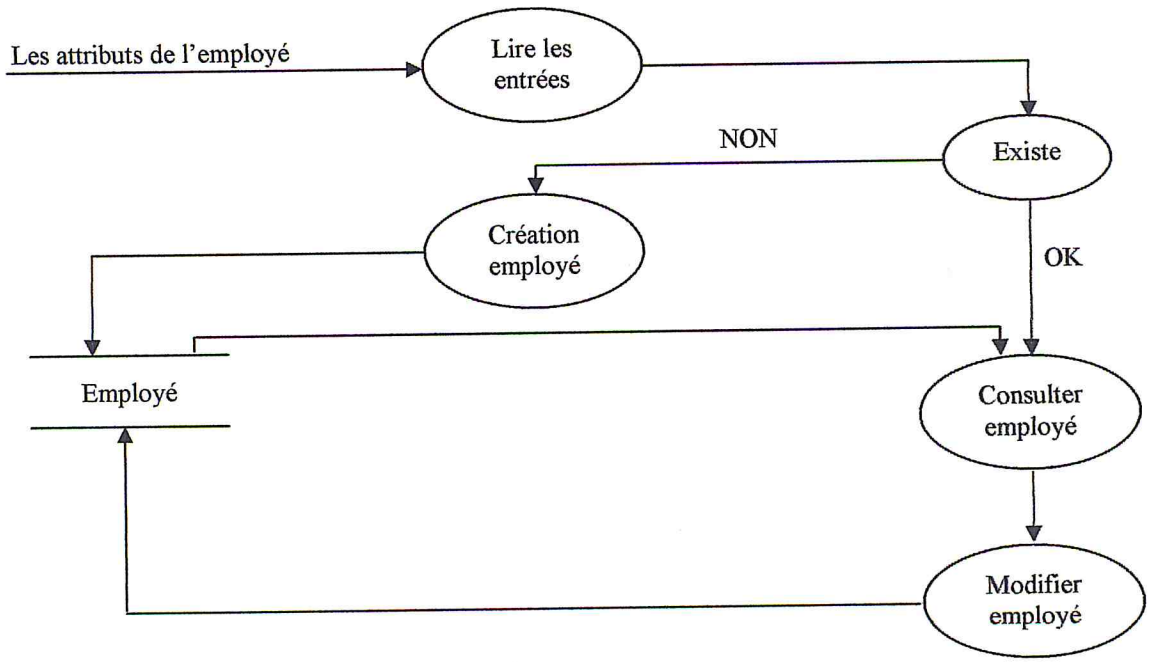
3.1.2. Diagramme à flots de données du traitement «Affectation d'un nouveau matériel»



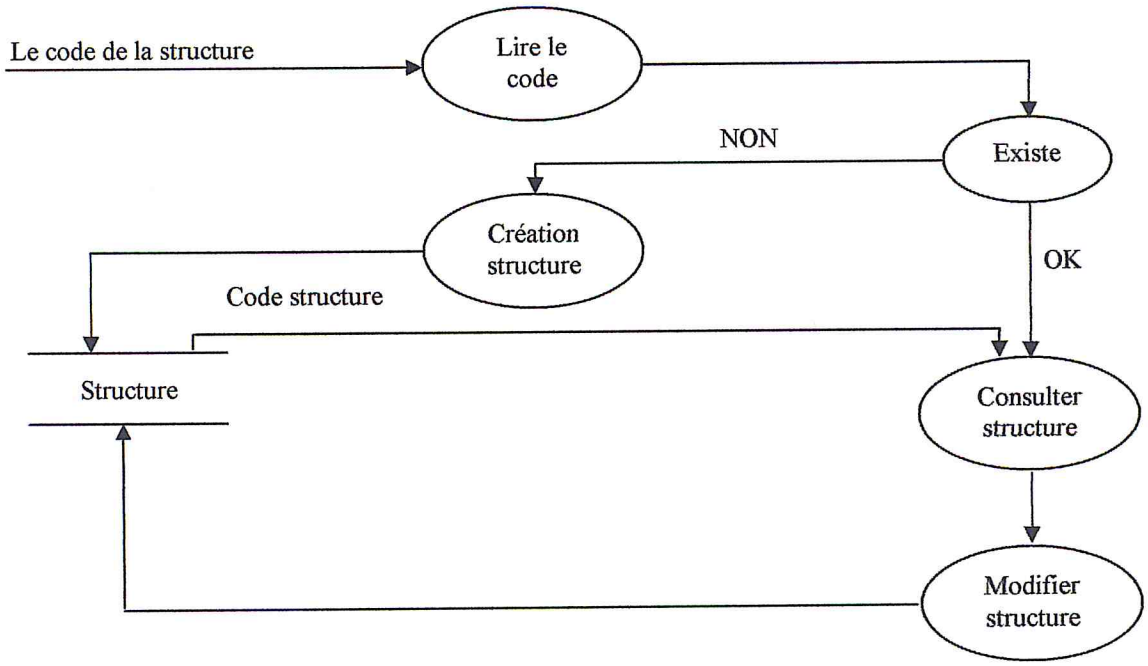
3.1.3. Diagramme à flots de données du traitement « Restructuration d'un étage »



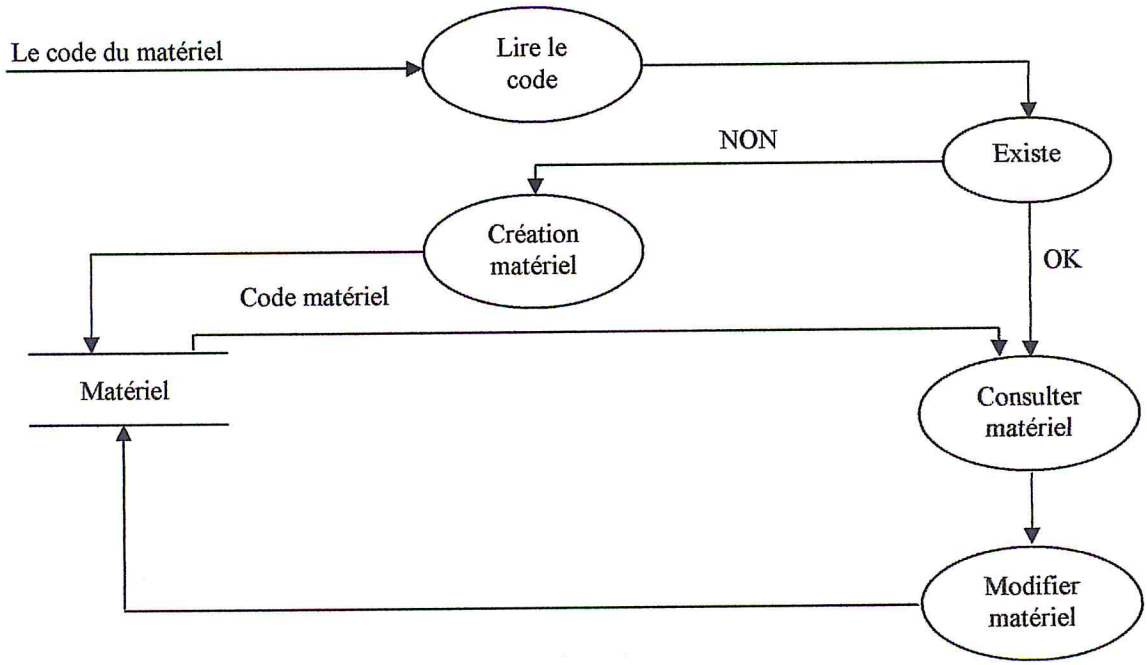
3.1.4. Diagramme à flots de données du traitement « Ajout et consultation employé »



3.1.5. Diagramme à flots de données du traitement « Ajout et consultation structure »



3.1.6. Diagramme à flots de données du traitement « Ajout et consultation matériel »



3.2. Fonctions communes à tous les diagrammes

Fonction Existe

Permet de rechercher l'objet afin de vérifier son existence.
Si l'objet existe, l'administrateur pourra éventuellement le consulter.
Sinon, on procède à sa création.

Fonction Création

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

Fonction Consultation

Permet de lire les données des classes et de les consulter si le contrôle de l'existence de l'objet est positif.

Fonction Modifier

Lorsqu'une information fait l'objet d'un changement, l'administrateur doit noter les rectifications nécessaires sur l'objet d'une classe donnée (modifier les attributs de l'objet ou supprimer un objet).

Fonction Réception

C'est-à-dire recevoir l'ordre, une demande, une affectation, un ajout, une suppression …, par un service (une sous structure {organisme}) concerné, traitant ces fonctions.

Fonction Contrôle

Le service de consultation d'une structure vérifie ce qui vient d'être réceptionné et demande plus d'informations aux ressources humaines.

Fonction Mise à jour

Confirmer les changements et mettre à jour la base de données.

Partie 2 :

Conception de Système

II. CONCEPTION DE SYSTÈME :

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

1. Règles de passage entre modèle objet et modèle tables relationnelles

Représentation des classes d'objets 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.
- Une association un à plusieurs est représentée 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 les objets 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 parfois à promouvoir une association n-aires en une classe.
- Une association qualifiée se représente 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 represente chaque super-classe et chaque sous-classe par une table.
- S'il n'y a pas de table 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 apporte tous les attributs des sous-classes dans la super-classe.

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 se présente comme suit :

Modèle objet	**Modèle de la table**		
Classe Employé	**Table Employé**		
	Nom attribut	Nulle ?	Type
Mat-empl	Mat-empl	Non	ID
Code-str	Code-str	Non	ID
Nom-empl	Nom-empl	Non	A
Prénom-empl	Prénom-empl	Non	A
Nom-j-fille	Nom-j-fille	Non	A
Date-nais-empl	Date-nais-empl	Non	Date
Lieu-nais-empl	Lieu-nais-empl	Non	A
Sexe	Sexe	Non	A
NAT	NAT	Non	A
ADR-empl	ADR-empl	Non	A
NS	NS	Non	N
Fct	Fct	Non	A
SF	SF	Non	A
Nbr-enf	Nbr-enf	Non	N

Modèle objet	**Modèle de la table**		
Classe Structure	**Table Structure**		
	Nom attribut	Nulle ?	Type
Code-str	Code-str	Non	ID
Design-str	Design-str	Non	A
Miss_str	Miss_str	Non	A
Nbre-empl	Nbre-empl	Non	N

Modèle objet	**Modèle de la table**		
Classe Matériel	**Table Matériel**		
	Nom attribut	Nulle ?	Type
Code-mat	Code-mat	Non	ID
Num-bur	Num-bur	Non	ID
Design-mat	Design-mat	Non	A
Type-mat	Type-mat	Non	A
Config	Config	Non	AN

Modèle objet	**Modèle de la table**		
Classe Bureau	**Table Bureau**		
	Nom attribut	Nulle ?	Type
Num-bur	Num-bur	Non	ID
Num-zone	Num-zone	Non	ID
Surface	Surface	Non	N
Lign-reseau	Lign-reseau	Non	AN

Modèle objet	**Modèle de la table**		
Classe Niveau	**Table Niveau**		
	Nom attribut	Nulle ?	Type
Num-niv	Num-niv	Non	ID
Num-bloc	Num-bloc	Non	ID

Modèle objet	**Modèle de la table**		
Classe Bloc	**Table Bloc**		
	Nom attribut	Nulle ?	Type
Num-bloc	Num-bloc	Non	ID
Code-str	Code-str	Non	ID
Nbre-zone	Nbre-zone	Non	N

Modèle objet	**Modèle de la table**		
Classe Zone	**Table Zone**		
	Nom attribut	Nulle ?	Type
Num-zone	Num-zone	Non	ID
Num-bloc	Num-bloc	Non	ID
Nbre-bur	Nbre-bur	Non	N

Modèle objet	**Modèle de la table**		
Classe Ligne Téléphonique	**Table Ligne Téléphonique**		
	Nom attribut	Nulle ?	Type
Num-poste	Num-poste	Non	ID

Partie 3 :

Conception des Objets

III. CONCEPTION DES OBJETS

Chaque table de 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 déduites des actions du modèle dynamique ou des traitements du modèle fonctionnel et en respectant les règles de passage énoncées précédemment. La conception des objets du système est définie par les prototypes des classes suivantes :

Classe : Table Employé	
Attributs	**Méthodes**
Mat-empl Code-str Nom-empl Prénom-empl Nom-j-fille Date-nais-empl Lieu-nais-empl Sexe NAT ADR-empl NS Fct SF Nbr-enf	Création employé Modifier employé Consulter employé Existe employé

Classe : Table Structure	
Attributs	**Méthodes**
Code-str Design-str Miss_str Nbre-empl	Création structure Modifier structure Consulter structure

Classe : Table Ligne Téléphonique	
Attributs	**Méthodes**
Num-poste	Création ligne téléphonique Modifier ligne téléphonique Consulter ligne téléphonique Existe ligne téléphonique

Classe : Bureau	
Attributs	**Méthodes**
Num-bur Num-zone Surface Lign-reseau	Consulter bureau Modifier bureau Existe bureau

Classe : Table Matériel	
Attributs	**Méthodes**
Code -mat Num-bur Design-mat Type-mat Config	Création matériel Modifier matériel Consulter matériel Existe matériel

Classe : Bloc	
Attributs	**Méthodes**
Num-Bloc Code-str Nbre-zone	Consultation bloc

Classe : Niveau	
Attributs	**Méthodes**
Num-Niveau Num-bloc	Consultation niveau

Classe : Zone	
Attributs	**Méthodes**
Num-zone Num-bloc Nbre-bur	Consultation zone

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 VI :

Implémentation et Mise en oeuvre

VI.1. Introduction

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.

L'ensemble de ces résultats a été soumis à l'approbation de la direction SPE.

Les résultats de cette étude feront l'objet de projection sur machine, selon l'outil choisi pour le niveau physique et permettront de rendre la solution opérationnelle.

VI.2. Environnement Technique de Développement

2.1. Présentation des langages de programmation utilisés

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

2.1.1. Le langage SQL

Pour la mise en place et la gestion de la base de données, le SGBD utilisé est le SQL, qui est un produit Microsoft, donc parfaitement compatible avec la plate forme choisie, de plus il est entièrement conçu pour le web.

Le SQL est un langage de requête structuré (**S**tructured **Q**uery **L**anguage) destiné à communiquer avec des bases de données relationnelles implémentées dans des SGBDR (Système de **G**estion de **B**ases de **D**onnées **R**elationnelles).

Le langage SQL permet d'effectuer diverses opérations comme la création, l'extraction, la modification, la suppression, la fusion, etc., sur des collections de données, par l'intermédiaire d'instructions particulières appelées des commandes, souvent assistées d'ailleurs par des clauses ou des options.

2.1.2. Le langage C#

Le langage C# créé par Microsoft dans le but de concurrencer Java, est devenu un standard approuvé par la spécification ECMA-334 de l'association européenne ECMA (European Computer Manufacturer's Association) en décembre 2001.

Le langage C# permet de développer des applications Windows et Web en se basant sur l'architecture .NET.

Ce langage a été conçu dans le but d'être indépendant de la plate-forme et de l'environnement d'exécution. Normalement, le compilateur C# devrait pouvoir fonctionner partout où les types et fonctionnalités spécifiés dans l'ECMA-334 sont correctement prises en charge par l'environnement d'exécution.

Actuellement, l'environnement d'exécution du langage commun (CLR : Common Language Runtime) de la technologie .NET est évidemment, le domaine de prédilection du langage C#.

Le mécanisme de traitement du code source écrit en C# par la plate forme Microsoft.NET, produit lors de la phase de compilation, un code binaire particulier dénommé MSIL (MicroSoft Intermediate Language), lequel sera interprété ensuite par le CLR chargé de la transformation en instructions exécutable par la machine.

La technologie .NET met également à la disposition du programmeur de nombreuses classes du .NET Framework, comprenant une pléthore de méthodes et propriétés exploitables dans des programmes C#.

Le langage C# est une savante combinaison des langages C, C++ et Java en expurgeant la plupart des sources d'erreurs et en améliorant certaines fonctionnalités.

La gestion de la mémoire par l'utilisation des pointeurs a été maintenue tout en veillant à y intégrer un dispositif d'appel explicite par l'utilisation de mot-clés spécifiques.

Le concept d'héritage multiple dans la programmation orientée objet a été abandonné au profit de l'utilisation d'interfaces.

Le langage C# permet une programmation orientée objet simple et moderne. Dérivé des langages de programmation C et C++, C# semblera familier aux habitués du développement en C, C++ et Java et ne sera sans aucun doute, pas plus difficile à apprendre pour les autres.

Effectivement, le langage C# ne comporte que 77 mots-clés. De plus, la programmation en C# se révèle intuitive, familière et moins verbeuse produisant du code se démarquant par sa lisibilité et sa concision.

2.2. Présentation des logiciels utilisés

2.2.1. MapInfo

MapInfo est un SIG généraliste bureautique typique. Il permet de sortir très facilement toutes sortes d'analyses thématiques. Il autorise l'utilisateur à ouvrir des fichiers EXCEL, à ouvrir et à modifier des fichiers ACCESS, à travailler sur des données ORACLE... de manière transparente. (Conception de notre base de données géographique).

2.2.2. MapXtreme

MapXtreme permet de créer des applications intégrant une dimension cartographique aussi bien en environnement monoposte, client/serveur que Web.

2.2.3. CorelDraw

La Suite graphique CorelDRAW® 12 innove en offrant des outils intelligents qui réduisent les étapes de la conception, simplifient le reciblage des graphismes, accélèrent la révision des projets, aident à mieux positionner textes et objets et optimisent les mises en page.

2.3. Implémentation

Notre base de données est construite par le SQL Server.

2.3.1. SQL Server

C'est un système relationnel de gestion de base de données (SGBDR) multi-utilisateurs à haute performance tournant sous le système d'exploitation Windows NT, il est conçu pour être exploité sur un ordinateur serveur et être le composant serveur dans un environnement Client/Serveur.

SQL Server exécute des fonctions SGBD telles que :

- La manipulation des données.
- La sécurité.
- La vérification d'intégrité.
- La sauvegarde et la restitution des données.

2.3.2. Le serveur web IIS

Dans ces dernières années, la possibilité de pouvoir connecter un système d'exploitation au réseau Internet est devenue déterminante. Pour cela, Microsoft a inclus un serveur d'informations Internet dénommé IIS ; il tourne aussi bien sous MS-Windows et Unix.

2.4. Sécurité de l'application

Le domaine de la sécurité est une notion très importante dans la programmation web, dans ce cas la sécurité est assurée par :

La sécurité de Windows NT :

Il existe plusieurs moyens d'améliorer la sécurité d'un ordinateur qui publie des informations sur un réseau intranet ou sur internet.
Windows NT offre des fonctionnalités de sécurité frontale pour le site Web, notamment l'authentification et les autorisations Web.

La sécurité des pages

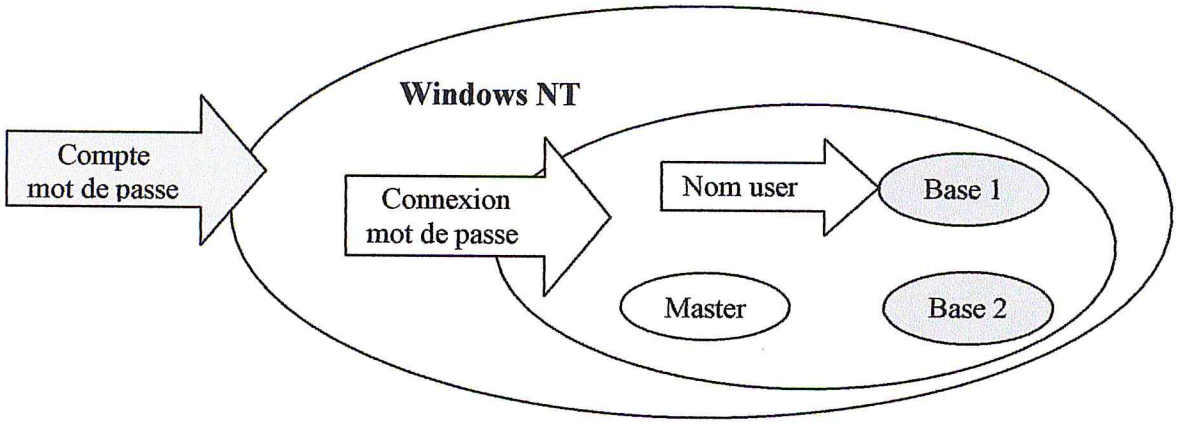
On sécurise certaines pages par l'authentification des utilisateurs, afin de restreindre l'ensemble des utilisateurs qui peuvent accéder à des parties spécifiques du site.

La sécurité SQL Server

SQL Server offre divers niveaux de protection des données qui lui sont confiées. Il s'agit tout d'abord de protéger l'accès au serveur. Ensuite, en fonction de son profil, l'utilisateur aura ou non l'accès à certaines bases de données. Enfin, à l'intérieur d'une base de données, il n'aura pas forcément le loisir d'accéder ou de modifier les informations comme il le souhaite.

Il est préférable de donner à un utilisateur un compte sur le serveur selon la configuration du serveur NT. Le SQL Server ne travaillera qu'a partir de la navigation d'ouverture de session effectuée par le Windows NT.

Du moment que l'utilisateur possède un compte sur le serveur ou le domaine NT, il devra posséder un nom de connexion pour ouvrir une session sur SQL Server. Ce nom de connexion sera généralement mis en correspondance avec un nom d'utilisateur dans la base à la quelle il souhaite accéder. Enfin, tout ne lui est pas forcément possible dans la base.

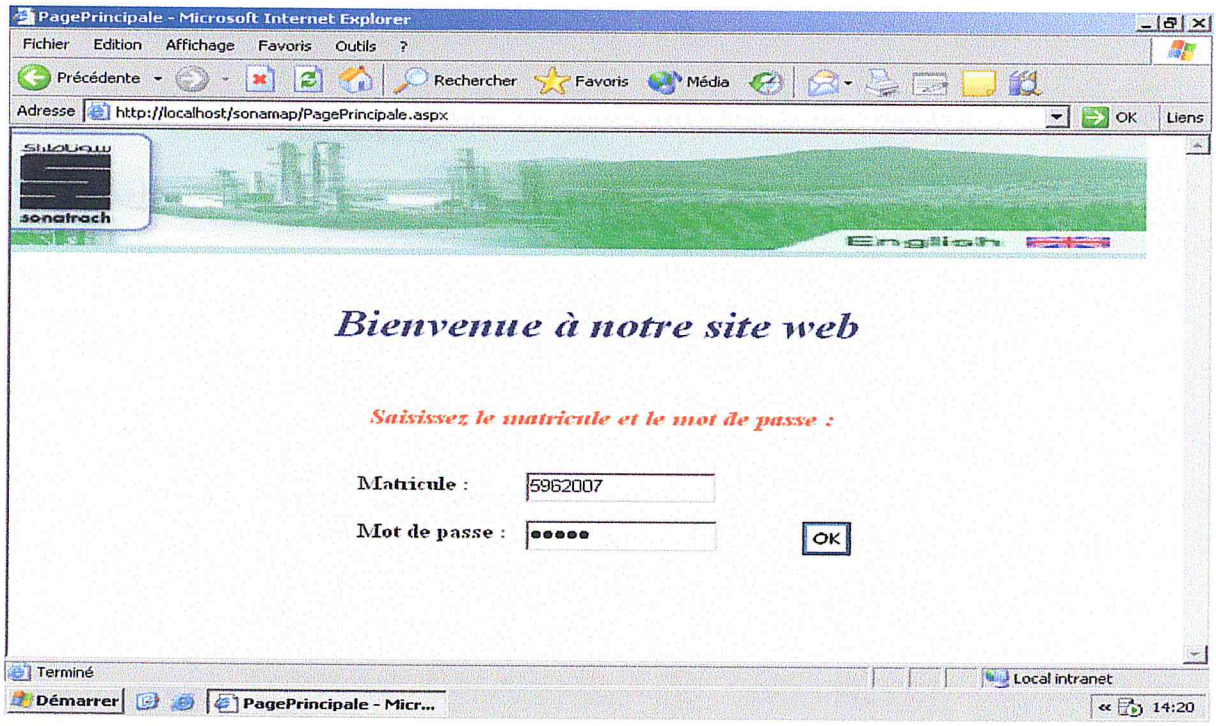


Architecture de l'accès au SQL Serveur.

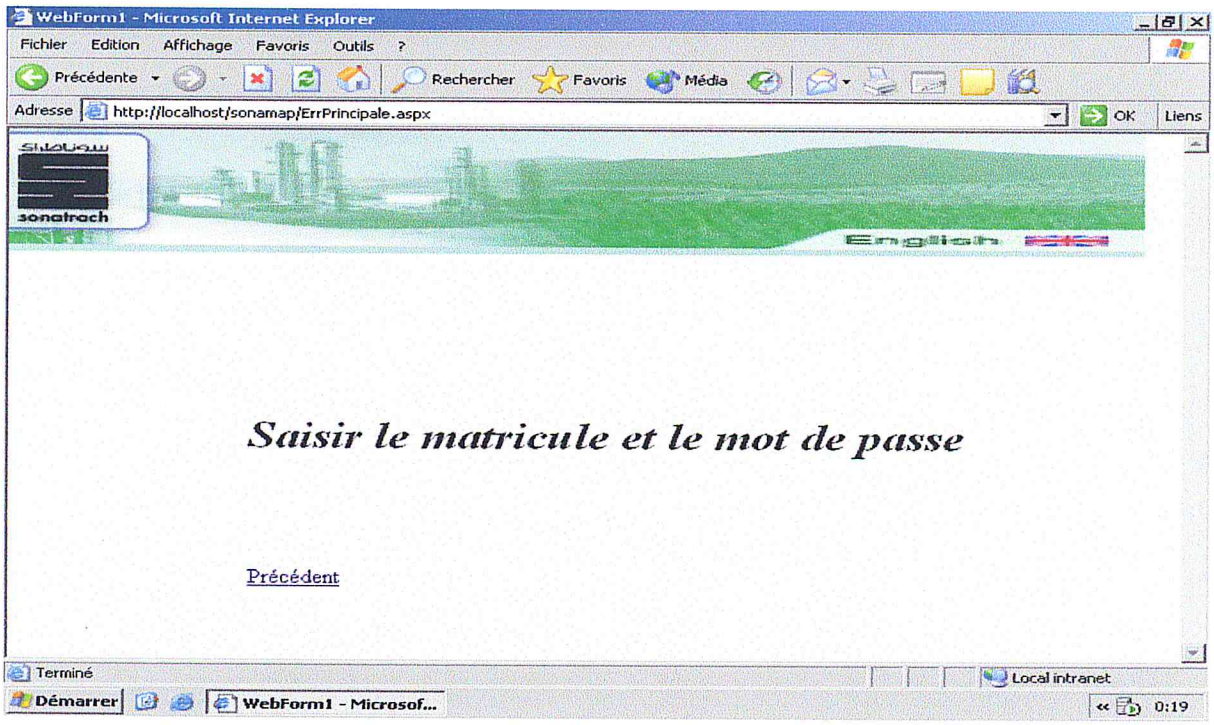
2.5. Test de l'application :

Notre application est constituée d'une suite de pages statiques.
Le visiteur du site peut faire différentes opérations de consultation ou de mise à jour sur les données de la base, selon ses droits d'accès.

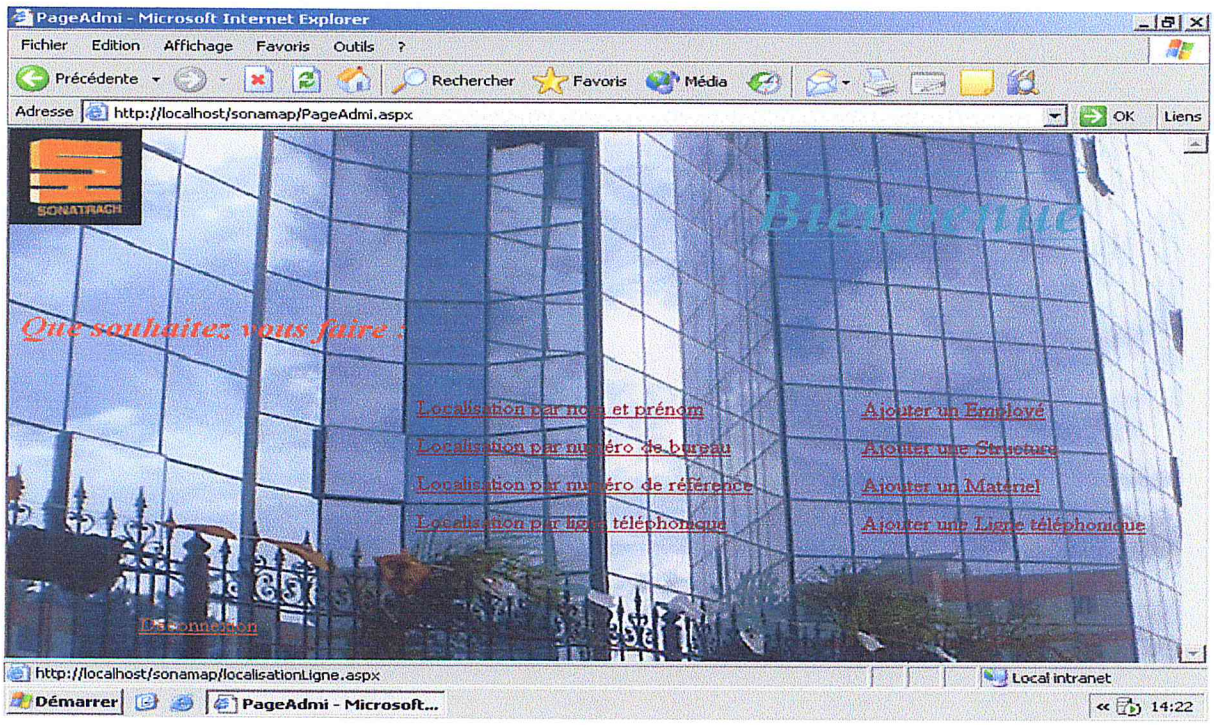
La première page est une page d'accueil, où l'utilisateur doit saisir son matricule et son mot de passe. l'accès au logiciel dépend du type de l'utilisateur, simple employé, personnel de la Sonatrach, ou l'administrateur du logiciel.



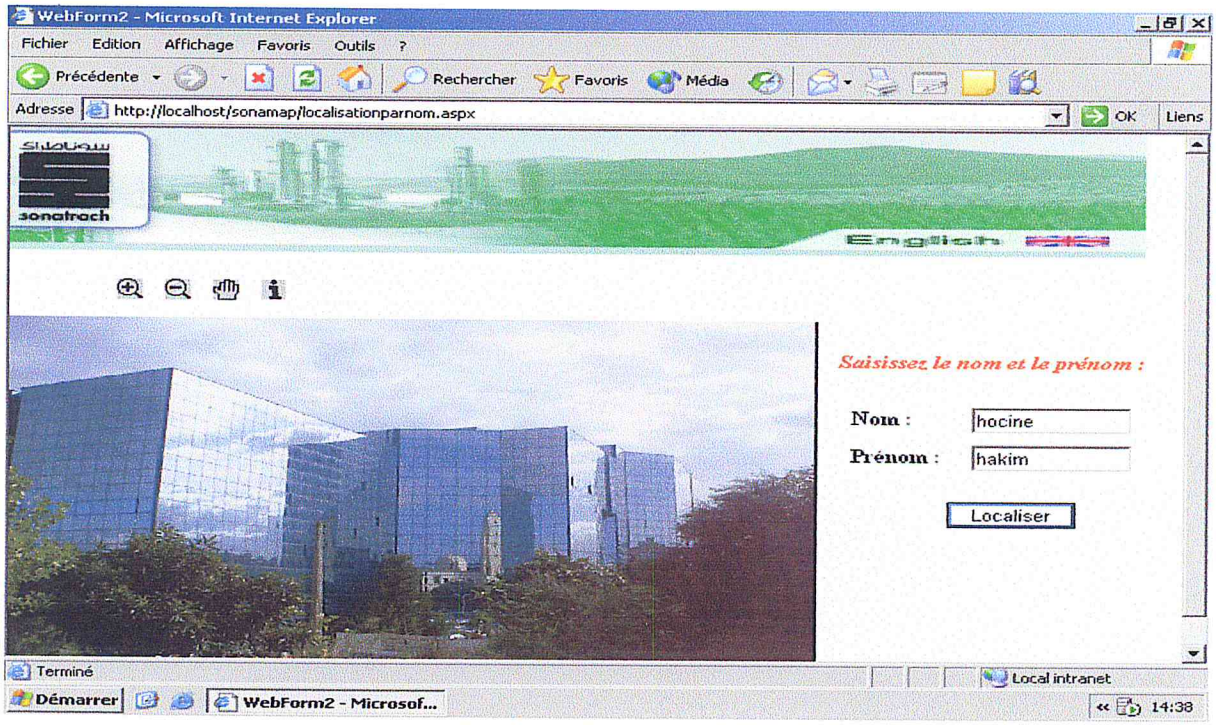
Après la saisie, un programme s'occupe de vérifier la validation des informations en entrée, en cas d'erreur, un message s'affiche pour l'indiquer :



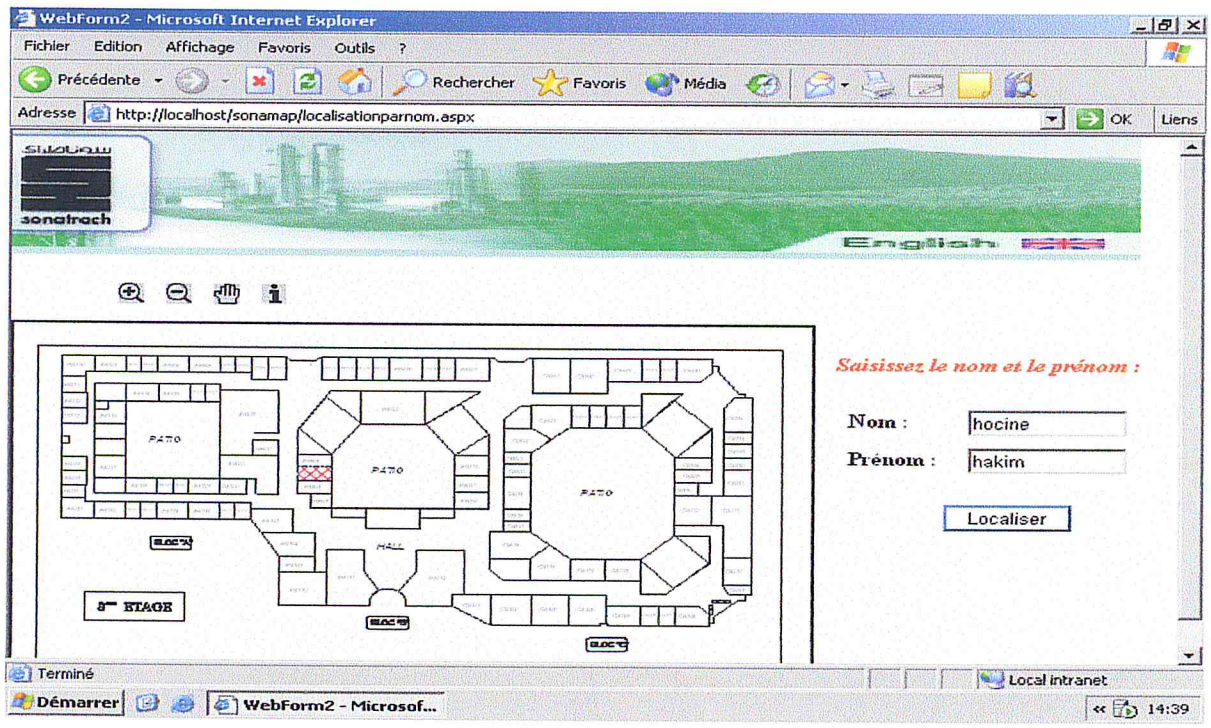
Si les données sont valides, l'administrateur aura une page contenant toutes les opérations qu'il peut effectuées :



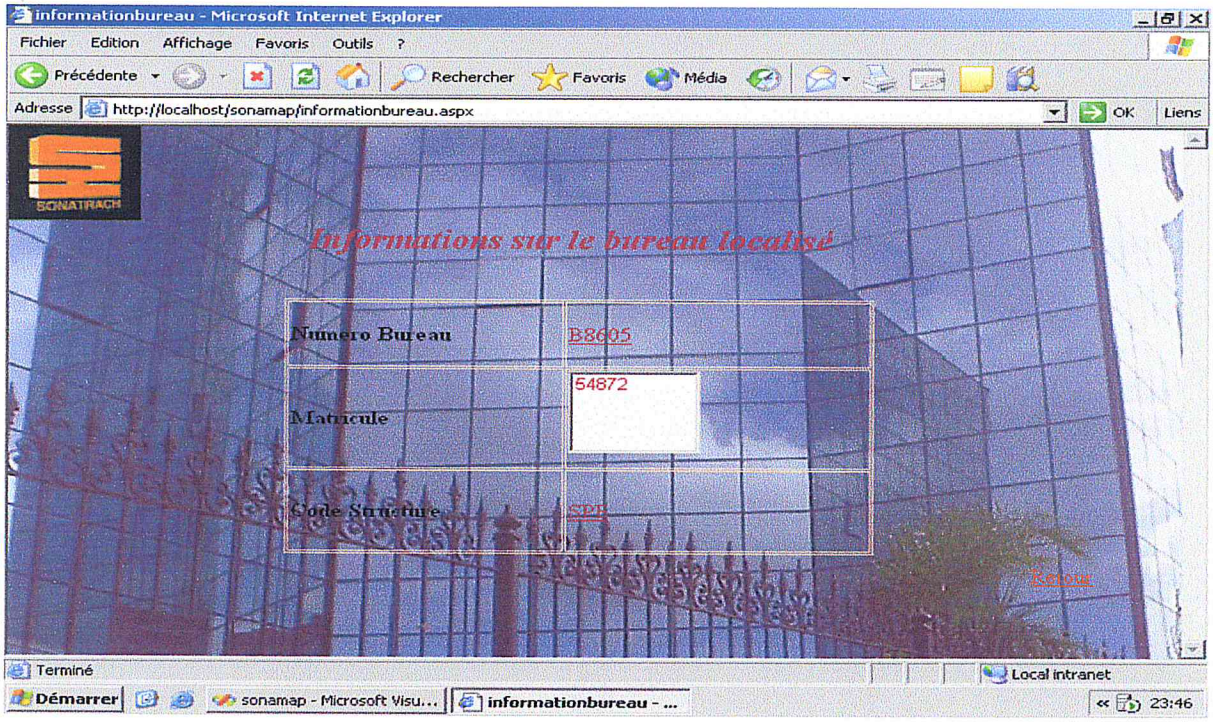
En choisissant de faire une localisation par nom et prénom, par exemple; on aura la page suivante :



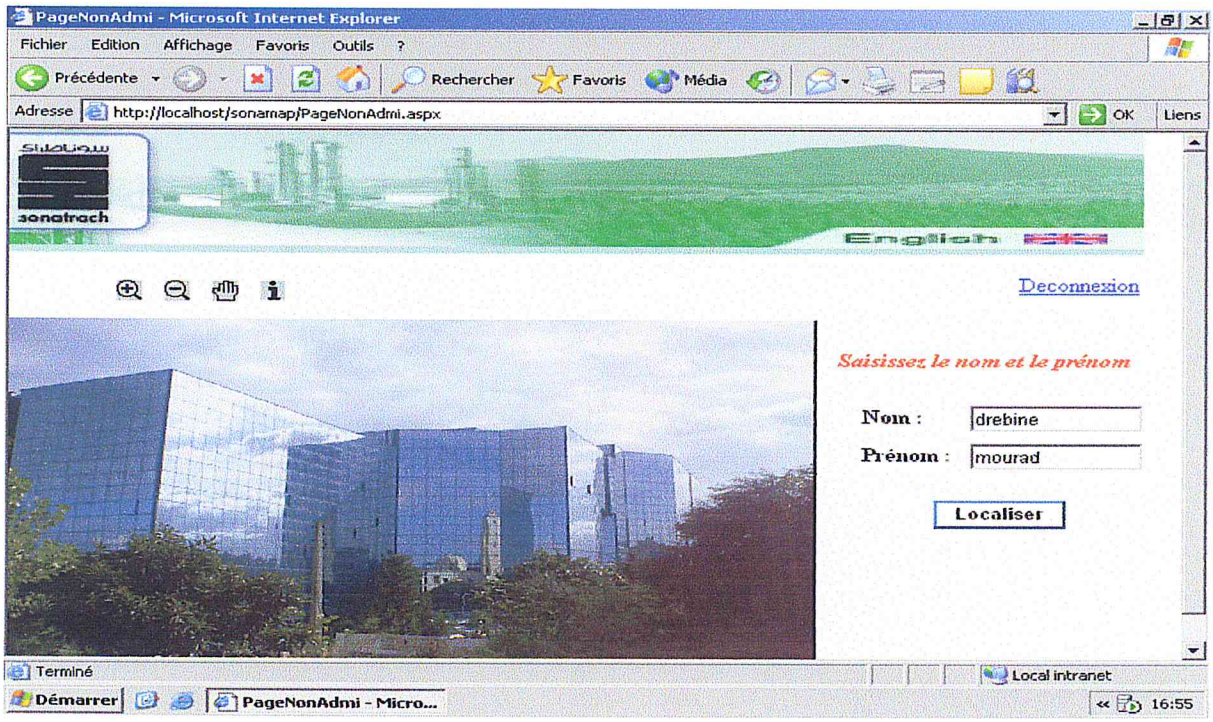
Après la saisie du nom et du prénom et le lancement de la recherche, le plan de l'étage correspondant apparaît avec spécification du bureau de la personne recherchée :



En cliquant sur le bureau correspondant, on aura une page avec toutes les informations concernant le bureau avec la possibilité de modifier ou de supprimer l'information de notre choix :



L'accès d'un simple employé ou personnel de la Sonatrach, offre seulement la possibilité d'effectuer une localisation par nom et prénom en accédant à la page suivante :



VI.3. Conclusion

Dans cette phase nous sommes arrivés à caractériser le système en se basant sur la conception réalisée dans les chapitres qui ont précédé et en utilisant les outils de développement tels que : SQL Server, Microsoft Visuel Studio .Net, MapXtreme, ...etc.

Conclusion et Perspectives

CONCLUSION

L'activité de la Direction Générale de SONATRACH a fait l'objet d'une analyse qui a permis de déceler les besoins liés à la localisation géographique et la difficulté de l'exploitation des données au moment opportun.

Par ailleurs, la définition des objectifs attendus a permis de dégager les orientations et les contours de la solution recherchée dont la conception d'un logiciel a constitué l'objet précis de notre projet.

Dans le cadre de la réalisation de ce projet, une approche méthodologique a été utilisée. Ainsi nous avons opté pour la méthode OMT. La base de données a été crée sous l'environnement WINDOWS sous le SGBD SQL Serveur 2000.

D'autre part, ce stage a été d'un grand rapport car il nous a permis d'aborder l'analyse et la spécification d'un système complexe réel, la mise en pratique des connaissances théoriques nous a aidés à compléter, initier et à maîtriser une grande partie de nouveaux outils mis à notre disposition à savoir :

- SQL Serveur 2000
- MapInfo
- MapXtreme
- Microsoft Visuel Studio.net
- CorelDraw
- Serveur IIS.

L'ensemble des résultats énoncés précédemment a été soumis à l'approbation de la direction générale.

Néanmoins, ces résultats devront être placés dans le contexte de cette étude, qui été marquée par de contraintes de temps externes au projet et par le manque d'expérience. Aussi, les modèles resteront ouverts à tous apports ou suggestions induites par le lecteur de ce document et ils pourront être pris en considération pour les mises à jours ultérieurs.

En effet, le projet pourrait bénéficier des extensions liées aux interfaces avec les systèmes qui existent déjà et ceux qui devraient être conçus et cela en plus des adaptations nécessaires à son implémentation dans son contexte réel.

Bibliographie

BIBLIOGRAPHIE

Ouvrages

[BOU 94] : BOUZEGHOUD Mokrane, George Gardarin & Patrick Valduriez
Objects : Concepts – Langages – Base de données – Méthodes – Interfaces.

[DAO 03] : DAO H., 2003, « Système d'information géographique, notes de cours », Département de géographie, Université de Genève.

[DID 90] : DIDIER M., 1990, « Utilité et valeur de l'information géographique», Etude du CNIG, Edition ECONOMICA, 256 p.

[DJE, BEN 99] : DJEZZAR F.Z, BENAMARA F., 1999, « conception et réalisation d'un système d'aide à la cécision pour la gestion des ressources naturelles », mémoire d'génieur d'état en informatique, université des sciences et technologie Houari Boumedienne.

[KAY 00] : KAYADJANIAN M., 2000, « Système d'information géographique », CESD, Luxembourg.

[MAR 02] : MARMONIER P., 2002, « L'information géographique », école national des science géographique (ENSG)- CERSIG, champs sur marne.

[RUM 00] : RUMBAUGH J et all, 2000, « OMT : Modélisation et conception », Masson.

[RUM 95] : RUMBAUGH J, 1995, « Objet modelling technique ».

[TUF 97] : TUFFERY C., 1997, « Les SIG dans les entreprises », Edition Hermès, Paris.

[ZAO 96] : ZAOUI M., 1996, « Modélisation des données du réseau de gaz et contribution à sa gestion au moyen d'un SIG », thèse de magister en techniques spatiales, centre national de techniques spatiales, Alger.

[1] : Microsoft, Programmation avec Microsoft Visual Basic.net, 2003
Cours 2374A ref n X08-80353

[2] : Microsoft, Présentation de Microsoft ASP.NET, 2002
Cours N° 2680A ref n x08-84792

Mémoires

Mémoire pour l'obtention du diplôme d'ingénieur d'état en génie informatique :
Thème : Conception et réalisation du système d'information global de la direction Information & Technologie de BROWN & ROOT – CONDOR, réalisé par BEGGAR Ines et BOUFADENE Taous- Boumerdes 2002-2003.

Mémoire pour l'obtention du diplôme d'ingénieur d'état en informatique :
Thème : Réalisation d'une base de données des informations et données statiques énergétiques de la SONATRACH, via le web; réalisé par CHERIF Karima et RARRBO Fatma- USTHB 2002-2003.

Les sites web:

www.google.com
www.microapp.com
www.labo-dotnet.com
www.develmoppez.com
www.commentçamarche.com
http://cedric.babault.free.fr/rapport/node4.html
http://perso.wanadoo.fr/thl/TechnoToile/Formation/Cours/4Tiers.htm

Annexes

1. Introduction :

L'information géographique présente une importance particulière pour les différents acteurs de la société tels que les hommes politiques qui souhaitent avoir une analyse des données sociales de la population, les militaires qui veulent déployer leurs équipements, un livreur qui s'intéresse à l'itinéraire le plus court pour livrer les meubles aux clients, un commerçant qui veut savoir où il doit placer son prochain magasin pour assurer un bon niveau de vente, un pompier qui veut savoir qu'elle est le meilleur itinéraire pour éteindre le feu le plus rapidement possible, un biologiste qui veut étudier le déplacement des populations animales, les hommes de santé qui veulent déterminer les zones homogènes de besoins de santé ou bien qui veulent suivre l'extension d'une maladie ou la survenue d'une éventuelle épidémie…

Les systèmes d'information géographiques sont des outils d'analyse, de diagnostic, de gestion, de simulation et d'aide à la décision. Par leur fiabilité et leur efficacité à répondre au besoin, ils participent au développement économique, à l'aménagement du territoire, à la production de connaissance (recherche, formation…), à l'analyse environnementale….

Dans l'histoire des technologies de l'information, l'apparition des SIG est très récente. On peut situer la naissance des SIG vers la fin des années 70, cette date coïncide avec l'émergence des premiers logiciels informatiques capables d'exploiter des données localisées. A titre d'exemple on peut citer le cas du logiciel SIG le plus utilisé au monde, le logiciel ARC-INFO, qui a vu le jour grâce à des travaux de recherches entrepris dans une université américaine au début des années 70. Mais c'est au début des années 80 que le traitement des données localisées s'est étendu à d'autres secteurs que la recherche et l'utilisation des SIG a dépassé le cadre dans lequel on voulait la cantonner, essentiellement la cartographie.

Aujourd'hui, les SIG sont utilisés dans presque tous les domaines et touchent à tous les secteurs d'activités et le nombre d'utilisateurs ne cesse de croître.

On a essayé à travers le présent chapitre, de présenter les principes et les notions de base des systèmes d'information géographiques.

2. Définition :

Depuis leur apparition, les SIG ont fait l'objet dans la littérature de plusieurs définitions, parmi celles qui ont eu un réel succès on peut citer :

"Un SIG est un système de gestion de base de données pour la saisie, le stockage, l'extraction, l'interrogation, l'analyse et l'affichage des données localisées" **[POR 92 in TUF 97].**

Une autre définition qui est tournée vers les besoins des décideurs est donnée par **DIDIER (90)** : "Un SIG est un ensemble de données repérées dans l'espace, structurées de façon à pouvoir en extraire des synthèses utiles à la décision".

Pour **ROUET (91)** : « les SIG utilisent des moyens informatiques pour stocker, consulter et manipuler les objets représentés sur les cartes ou les plans, ainsi que les informations qui leur sont directement ou indirectement attachées » **[TUF 97]**

Le Comité fédéral de coordination inter agences pour la cartographie numérique (1988) le définit comme étant un système informatique de matériels, de logiciels et de processus conçus pour permettre la collecte, la gestion, la manipulation, l'analyse, la modélisation et l'affichage de données à référence spatiale afin de résoudre des problèmes complexes d'aménagement et de gestion.

Le comité scientifique du colloque intégration de la photogrammétrie et de la télédétection dans les SIG SFPT (Strasbourg 1990) a adopté une autre définition : « Un SIG est un système informatique permettant, à partir de diverses sources, de rassembler et organiser, de gérer, d'analyser et de combiner, d'élaborer et de présenter des informations localisées géographiquement contribuant notamment à la gestion de l'espace »

Ainsi il ressort qu'un SIG est un système informatisé (matériels, méthodes et logiciels) permettant de gérer, d'organiser, d'analyser, d'élaborer, d'afficher et de présenter des données localisées géographiquement, structurées de façon à pouvoir en extraire commodément des synthèses utiles à la décision, à la gestion et à l'aménagement de l'espace.

3. Les principales disciplines impliquées :

Les SIG constituent une technologie multidisciplinaire qui intègre des principes, des méthodes et des technologies hérités de plusieurs disciplines traditionnelles, comme le démontre la **Figure 1** :

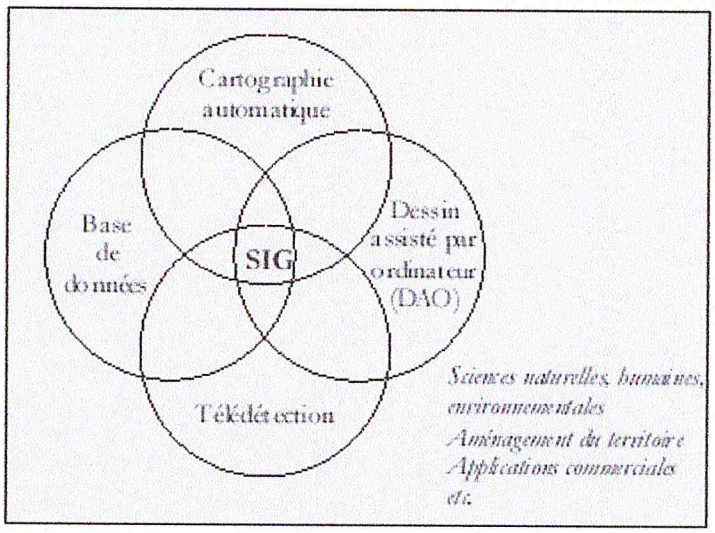


Fig. 1 : Principales disciplines à la croisée des SIG
[DAO 2003]

4. Fonctionnalités des SIG :

Les SIG sont généralement définis par 5 fonctionnalités principales (les 5 A):

- Acquisition : collecte et numérisation des données (calage, numérisation, importation et exportation de données)
- Archivage : Gestion et Stockage des données (fonction d'un SGBD).
- Affichage : visualisation de l'information géographique.
- Analyse : requête, modélisation et simulation.
- Abstraction : schéma conceptuel de données, dictionnaires et répertoires de données. **[MAR 2002][DAO, 2003]**

La **Figure 2** fait ressortir ces 5 fonctionnalités qu'on devrait retrouver dans chaque SIG :

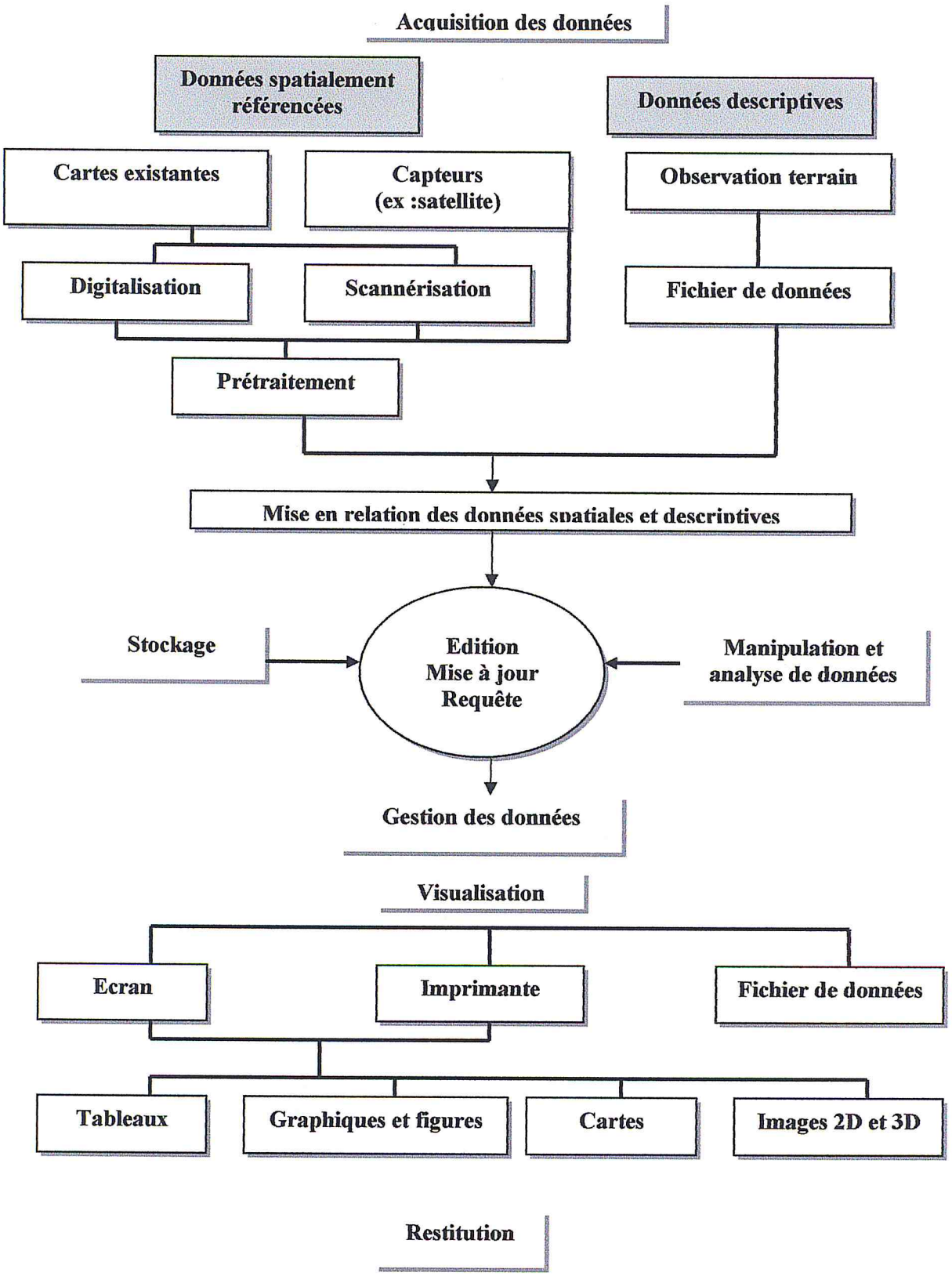


Fig. 2 : Principales fonctions d'un SIG
[DID 1990 in ZAO 1996]

5. Les questions auxquelles peuvent répondre les SIG :

Il existe plusieurs questions de base auxquelles un SIG doit pouvoir répondre, ce qui explique la multiplicité des domaines d'application de ce dernier. Un SIG doit pouvoir répondre à 5 grandes questions :

- **Où ?** répartition spatiale et localisation d'un objet ou d'un phénomène.
- **Quoi ?** que trouve-t-on à cet endroit ? Inventaire de tous les objets de l'espace concerné, identification et description des objets, superpositions, proximités.
- **Comment ?** relations existantes entre les objets ou phénomènes, analyse spatiale, problématique, optimisation.
- **Quand ?** à quel moment des changements sont intervenus ? Age, évolution d'un objet ou d'un phénomène, historique et actualisation des données, analyse temporelle.
- **Et si ?** Que se passerait-il si tel scénario d'évolution se produisait et quelles en seront les conséquences ? Projection dans l'avenir, simulation, études de projet ou d'impact **[DAO, 2003]**.

6. Domaines d'application du S.I.G :

Si l'on essaie de caractériser les questions auxquelles un S.I.G est censé pouvoir répondre, on est vite confronté à la multiplicité des domaines d'application possibles.

- aménagement du territoire : Schémas de Cohérence Territoriale (SCOT), Plans Locaux d'Urbanisme (PLU), choix de tracés routiers, autoroutiers ou ferroviaires, études d'impacts....
- gestion urbaine : gestion de la voirie, des réseaux de distribution, des espaces verts, du patrimoine, de la sécurité, simulation d'insertion de projets architecturaux....
- circulation et conduite automobile : choix d'itinéraires, suivi de flottes de véhicules, aide à la conduite assistée par ordinateur,
- agriculture : génie rural, gestion des ressources en eau, suivi et prévision des récoltes, gestion des forêts, aide à la mise en œuvre de la Politique Agricole Commune
- protection de l'environnement : définition des zones sensibles, suivi des évolutions, alerte aux pollutions, protection des paysages
- risques naturels et technologiques majeurs : définition et suivi des zones à risque, prévention de catastrophes, intervention en cas de sinistre, organisation des secours .

7. Les composantes d'un SIG : [KAY 2000]

Un système d'information géographique est constitué de 5 composants majeurs qui sont essentiels pour son bon fonctionnement :

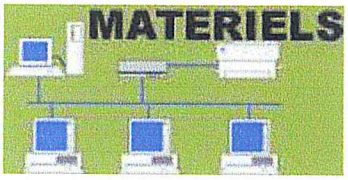
- **Matériels :** le SIG fonctionne aujourd'hui sur une gamme très diversifiée d'ordinateurs : des micro-ordinateurs (PC ou Mac), des stations de travail sous Unix et des serveurs de données, des ordinateurs de bureaux utilisés en autonomie ou selon des configurations en réseau, sans oublier les périphériques qui permettent d'assurer diverses fonctions (matériels d'acquisition, de stockage et de visualisation des données).

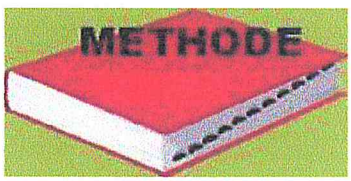
- **Logiciel :** les SIG fournissent les fonctions et les outils qui permettent de stocker, d'analyser et d'afficher ces données géographiques.

- **Données :** ce sont certainement les composantes les plus importantes d'un SIG. le SIG peut intégrer des données géométriques et des données attributaires. Cette association est l'une de ces fonctionnalités clé.

- **Utilisateurs :** un SIG étant avant tout un outil, ce sont les utilisateurs (qui dirigent, entretiennent et gèrent le système en élaborant des plans pour l'appliquer à des problèmes réels) qui lui donnent toute l'efficacité dont il peut être porteur.

- **Méthodes :** Le succès relatif à la mise en oeuvre et à l'utilisation d'un SIG ne peut se concrétiser sans l'application de méthodes, de règles et de procédures rigoureuses et cohérentes, adaptées à la stratégie et aux objectifs de l'organisation ou de l'entreprise.





8. Conclusion :

Un Système d'Information Géographique englobe les concepts suivants :

L'ensemble « Informations géographiques - Fonctionnalités » constitue l'outil SIG, qui permet d'apporter des solutions aux problèmes posés.

Créer un projet SIG est un investissement important, matériel et personnel. Il faut donc exprimer clairement les besoins et les objectifs. En général, les besoins et les objectifs correspondent aux solutions apportées par le SIG.

Les solutions peuvent être sous forme de :

- analyses thématiques, statistiques, spatiales,
- cartes,
- mise à jour de données,
- rassemblement de données

Pour justifier l'investissement fourni, le SIG doit être vivant. Il doit être fonctionnel, permet des analyses et évolue dans le temps, par la gestion de la mise à jour et de la qualité.

I- Présentation de la Plate-forme Microsoft .Net

I.1- La Plate-forme .NET

Cette section présente les grandes lignes de l'architecture de la Plate-forme .NET. La Plate-forme .NET se compose de plusieurs fonctionnalités et services de base, comme l'illustre la figure A.1, l'un des objectifs de cette nouvelle plate-forme est de simplifier le développement Web. **[1], [2]**

I.2- Technologie de base de la Plate-forme .NET

Les technologies de base qui composent la Plate-forme .NET sont les suivantes : **[1], [2]**

- .NET Framework :

  Cette technologie se fonde sur un nouveau Common Language Runtime. Celui-ci fournit un ensemble commun de services pour les projets crées avec Visuel Studio.NET, indépendamment du langage. Ces services fournissent des blocs modulaires de base pour les applications de tout type, utilisables à tous les niveaux des applications.
  Microsoft Visuel Basic, Microsoft Visuel C++, Microsoft Visuel C# et d'autres langages de programmation Microsoft ont été améliorés pour tirer profit de ces services.

- .NET Building Block Services :

  C'est un ensemble de services programmables distribués disponibles à la fois en ligne et hors connexion. Un service peut être appelé sur un ordinateur autonome non connecté à Internet, il peut également être fourni par un serveur local fonctionnant au sein d'une entreprise. .NET Building Block Services peut être utilisé à partir de n'importe quelle plate-forme prenant en charge SOAP. Au nombre de ces services, citons : les calendriers, les annuaires, la notification et la messagerie... etc.

- Visuel Studio .NET :

  Constitue un environnement de développement de haut niveau, destiné à la création d'applications sur le .NET Framwork. IL fournit des technologies clés afin de simplifier la création.

- .NET Entreprise Server :

  Les produits .NET Entreprise Server permettent une évolutivité, une fiabilité, une gestion de l'intégration. Ils offrent en outre un grand nombre de fonctionnalités par exemple : Microsoft SQL Server 2000, Microsoft Commerce Server 2000.

La figure suivante décrit les technologies de base de la Plate-forme Microsoft.NET

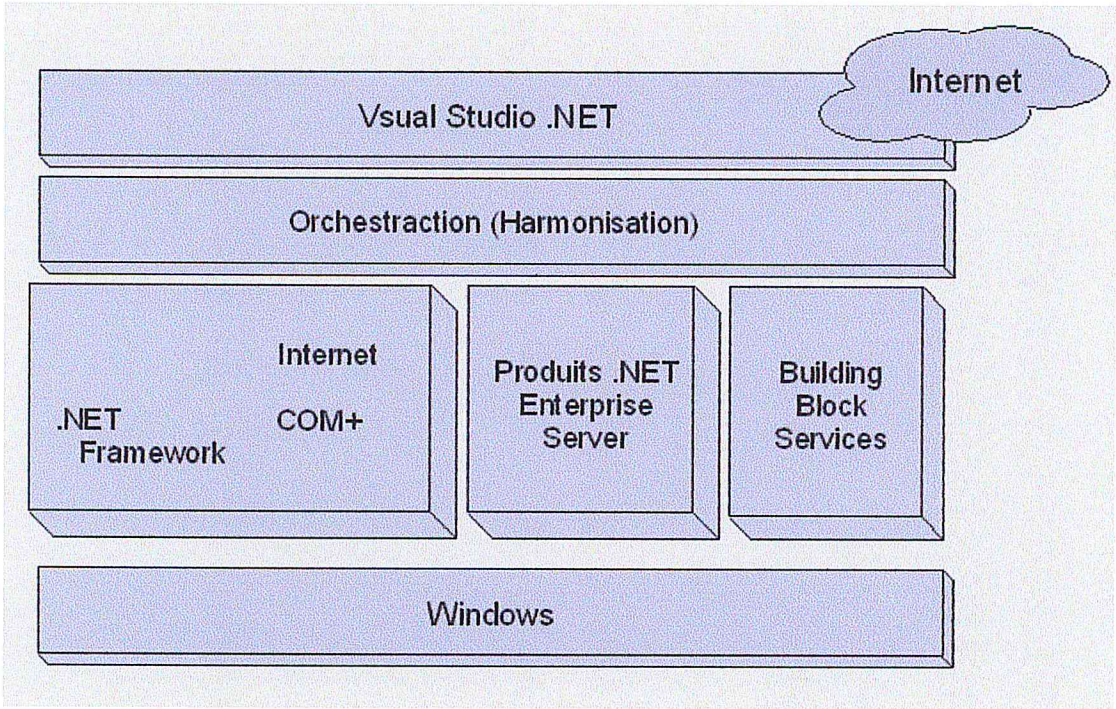


Figure A.1: *Plate forme Microsoft .NET*

I.3- Les avantages de la nouvelle technologie

La Plate-forme .NET offre les avantages suivants : **[1], [2]**

- Modèle de programmation cohérent et indépendant du langage, utilisable à tous les niveaux d'une application.
- Interobabilité parfaite entre technologies.
- Migration aisée à partir des technologies existantes.
- Prise en charge totale des technologies Internet fondées sur des standards et indépendantes des plate-forme, telles que http, XML et SOAP.
- Grâce au Common Language Runtime, tous les langages compatibles avec la Plate-forme .NET vont utiliser les mêmes fichiers d'exécution. Il n'est donc plus nécessaire de distribuer des bibliothèques d'exécution spécifiques à un seul langage, parce que les fichiers d'exécution .NET seront installés automatiquement dans les versions de Microsoft.

II- Présentation du .NET Framework

Le .NET Framework fournit tous les services communs nécessaire pour l'exécution de nos applications. Ces services sont disponibles dans tous les langages compatibles avec .NET grâce à la spécification CLS. **[1], [2]**

Cette figure décrit ces services :

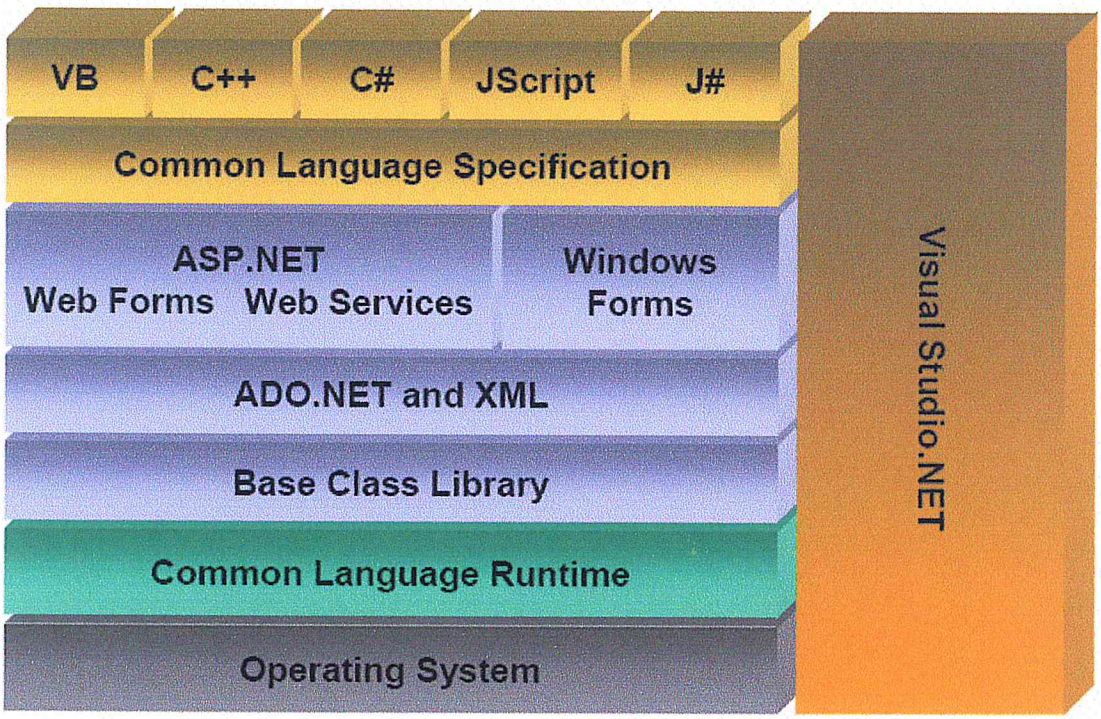


***Figure A.2** : Services communs pour l'exécution des applications.*

II.1- Création de composants dans le .NET Framework

Avant l'avènement de COM, les applications étaient des entités totalement séparées, sans aucune intégration ou très peu. Grâce à COM, nous pouvons intégrer des composants au sein d'une même application et à plusieurs, en exposant des interfaces communes. Dans le .NET Framework, les composants possèdent une base commune. Il n'est plus nécessaire d'écrire le code visant à permettre aux objets d'interagir directement les uns avec les autres.

Dans notre environnement, le .NET Framework prend totalement en charge les classes, l'héritage, les méthodes, les propriétés, le polymorphisme, les constructeurs et d'autres constructions orientées objet. **[1], [2]**

II.2- Spécification CLS (Common Language Specification)

La spécification CLS définit les standards communs que doivent respecter les langages et les développeurs pour que leurs composants et applications puissent être largement utilisés par d'autres langages compatibles avec le modèle .NET . Elle permet aux développeurs Visuel Basic .NET, Visuel C++ ou d'autres langages de créer des applications dans le cadre d'une équipe multi-langage, avec l'assurance que l'intégration des différents langages s'effectuera sans problème. CLS permet même aux développeurs Visuel Basic .NET ou Visuel C++ ...etc d'hériter de classes définies dans des langages différents. **[1], [2]**

II.3- Visuel Studio .NET

Dans le .NET Framework, Visuel Studio .NET fournit les outils servant au développement rapide d'applications. **[1], [2]**

II.4- Les Langages du .NET Framework

Cette section présente les langages que Microsoft fournit avec Visuel Studio .NET pour le .NET Framework. **[1], [2]**

- **Visuel Basic .NET :** Nouvelle version de Visuel Basic avec des innovations substantielles en terme de langage.
- **C# :** Il s'agit du premier langage moderne orienté composant.
- **Extensions C++ :** offre plus de puissance et de contrôle.
- **J# .NET :** est un langage pour les développeurs java qui souhaitent créer des applications et des services pour le .NET .
- **Langages tiers :** divers langages tiers prennent en charge le .NET : COBOL, Pascal, SmallTalk... etc.

III- Présentation des composants .NET Framework

Les composants du .NET Framework sont les suivants :

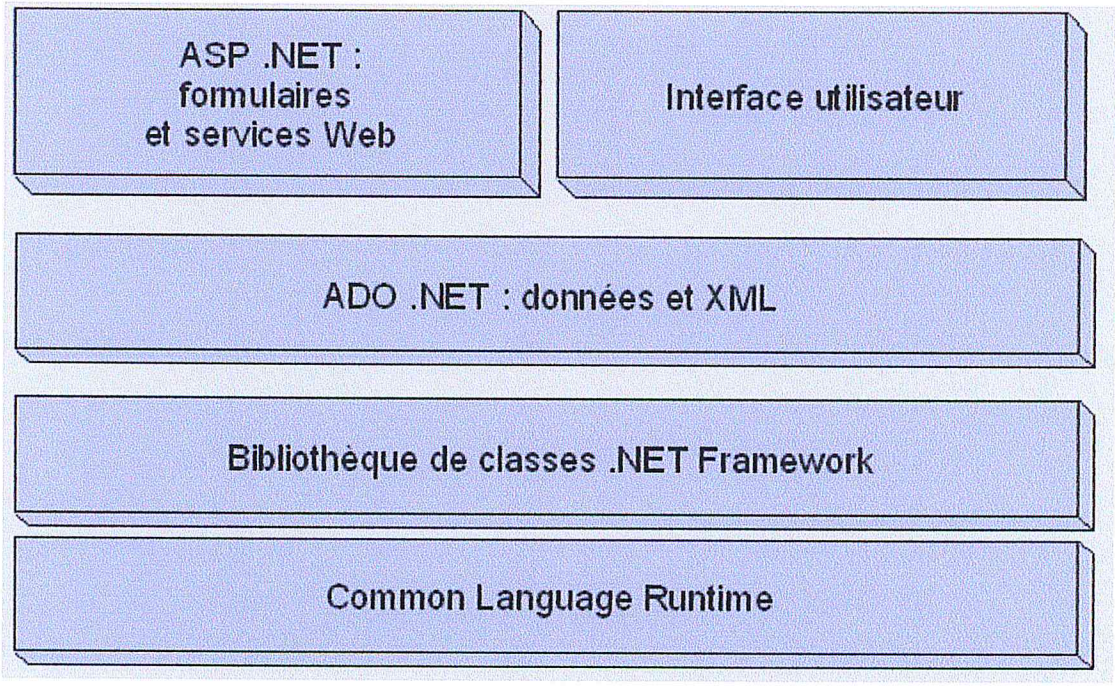


Figure A.3 : *Composants du .NET Framework.*

III.1- Common Langage Runtime

Il simplifie le développement d'applications, fournit un environnement d'exécution robuste et sécurisé, prend en charge plusieurs langages, simplifie le déploiement et la gestion des applications et offre un environnement géré.

III.1.1- Composants du Common Langage Runtime

Les fonctionnalités du Common Langage Runtime sont décrites dans la figure suivante :

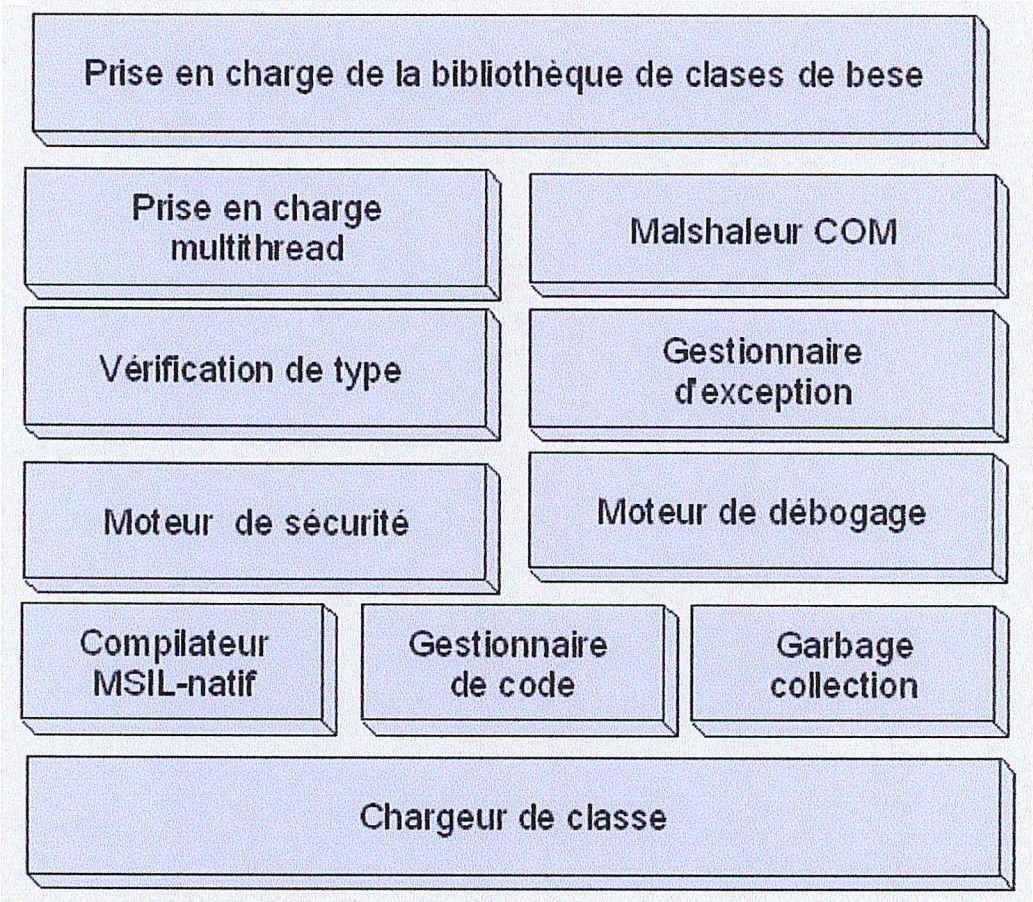


***Figure A.4** : Composants du Common Langage Runtime.*

- **Chargeur de classe :** charge en mémoire l'implémentation d'un type chargeable et le répare à l'exécution.
- **Compilateur MSIL - natif :** convertit MSIL en un code natif.
- **Gestionnaire de code :** gère l'exécution du code.
- **Garbage collection :** fournit une gestion automatique de la durée de vie de tous nos objets.
- **Moteur de sécurité :** fournit une sécurité par preuve, fondée sur l'origine du code en plus de l'utilisateur.
- **Moteur de débogage :** permet de déboguer l'application et de tracer l'exécution du code.
- **Vérification de type :** n'autorisera pas les conversions non sécurisées ou les variables non initialisées.
- **Gestionnaire d'exceptions :** fournit un traitement structuré des exceptions.
- **Prise en charge multithread :** fournit des classes et des interfaces qui permettent la programmation multithread.

- **Marchaleur COM :** fournit le marshaling à partir et à destination de COM.
- **Prise en charge de la bibliothèque de classes.NET Framework :** intègre du code au runtime qui prend en charge la bibliothèque de classes.

III.2- Bibliothèque de classes .NET Framework

Elle fournit de nombreuses nouvelles fonctionnalités puissantes du runtime et d'autres services essentiels de haut niveau via une hiérarchie d'objets qui s'appelle un espace de nom. **[1], [2]**

La figure suivante décrit ces espaces de nom :

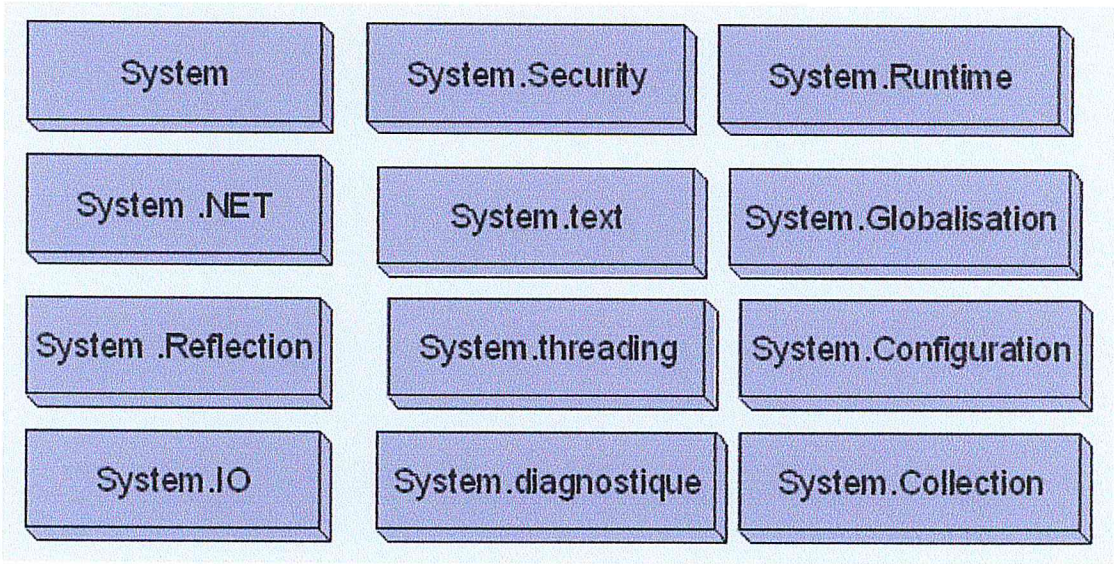


Figure A.5 Bibliothèque de classes .NET

- **System :** contient des classes fondamentales et de base qui définissent les types de données, les événements, les interfaces, les attributs… etc.
- **System.Collection :** fournit des listes, des tables et d'autres méthodes de regroupement de données.
- **System.IO :** il s'agit d'entrée /sortie et flux de fichiers.
- **System.NET :** fournit une prise en charge des sockets et de TCP/IP.

Pour plus d'informations, consulter la documentation du SDK Microsoft .NET Framework.

III.3- ADO .NET données et XML

ADO.NET est la nouvelle génération de la technologie ADO. Son but est l'amélioration du modèle de programmation déconnecté, ainsi elle est riche de XML. **[1], [2]**

- **System.Data :** comprend la classe **DataSet** qui représente des tables multiples et leurs relations.
- **System.XML :** il comprend un outil d'écriture et un analyseurXML.

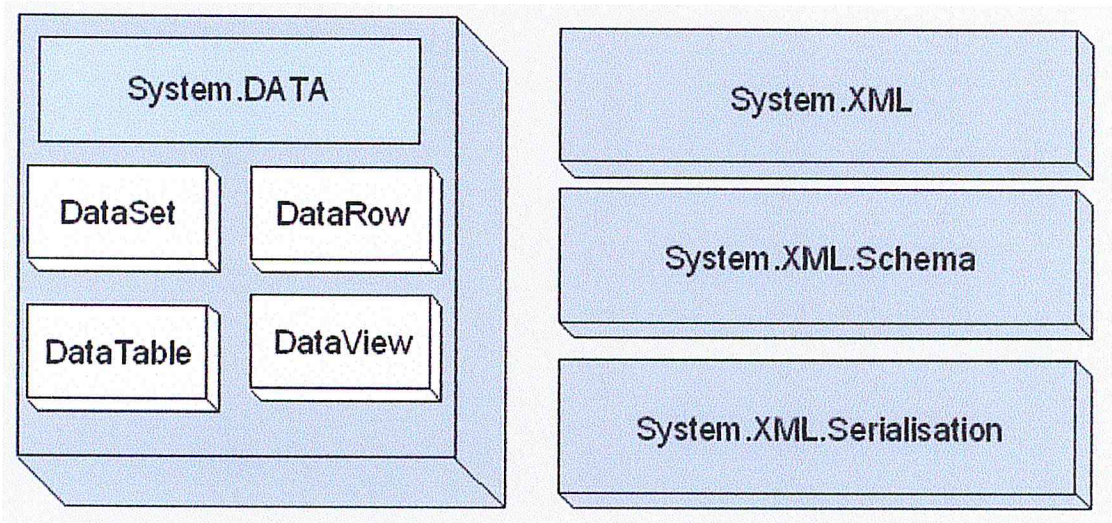


Figure A.6 : *ADO.NET*

III.4- ASP .NET formulaires et services Web (Active Server Pages)

ASP.NET est un cadre de programmation élaboré sur la base du Common Language Runtime et qui peut être employé sur un serveur pour créer des applications Web puissantes. Les formulaires ASP.NET sont des outils d'emploi pour la création d'interfaces utilisateur Web dynamiques. **[1], [2]**

- **System.windows.Forms** représente l'interface utilisateur coté client, tandis que **System.Drawing** représente la nouvelle génération de services GDI + (Graphic Device Interface Plus).
- Le Gestionnaire des services Internet, qui permet d'administrer les services Internet ;
- Le connecteur de base de données Internet (IDC, *Internet Database Connector*), qui sert à envoyer des demandes aux bases de données ;
- La gestion de l'Active Server Page (A.S.P.) de la technologie DCOM de MicroSoft, qui va servir à une programmation dynamique, et un accès aux Bases de données par l'utilisation du modèle Active Data Object (A.D.O.).
- Le Gestionnaire de clés, qui sert à installer les clés de sécurité SSL (*Secure Sockets Layer*).

MapInfo

1. Introduction

MapInfo est le leader mondial des Systèmes d'Information Géographique sur PC. Puissant et convivial, il permet de réaliser des analyses géographiques ad hoc. MapInfo Professional® est reconnu comme étant l'outil cartographique le plus puissant et le plus intuitif en environnement bureautique. Ses principales caractéristiques sont la connexion transparente aux bases de données relationnelles, la cartographie 3D et les outils pour la création de graphiques, de rapports et de pages HTML. MapInfo Professional peut être personnalisé et intégré à d'autres applications.

2. Le petit monde de MapInfo

MapInfo est un logiciel modulaire qui s'articule autour du logiciel MapInfo Professionnel® version 6.5. Ce logiciel peut aussi bien être fourni en version mono poste que multi utilisateurs accessible par réseau.

1. MapInfo Professional V6.5 est un outil de type Système d'Information Géographique qui sert à créer de l'information géographique, à traiter de l'information et à la cartographier. Pour simplement visualiser de l'information géographique au format MapInfo, le visualiseur MapInfo ProViewer v6.5 gratuit est suffisant.

En complément de MapInfo Professional®, il existe d'autres outils qui peuvent nous aider à régler des problèmes. Ces outils sont évoqués comme suit :

2. Vertical Mapper 3.0 (Vertical Mapper™) est un outil de création et d'exploitation de l'Information Géographique sous forme de grilles (Grid) assez puissant (MNT, exploitation d'images raster en relief...). Vertical Mapper™ est un logiciel diffusé par la société Marconi de type Plug-in, qui s'utilise avec l'environnement MapInfo Professional®. Ce logiciel n'existe qu'en langue anglaise.

3. ChronoMap® 2.1 et ChronoVia® - logiciels de la société française Magellan Ingénierie - qui sont conçus pour le calcul d'itinéraires, l'optimisation de tournées et la recherche de proximité (isochrones).

4. MapBasic® est le langage de programmation qui permet de personnaliser une application MapInfo. Par exemple, étendre ses fonctionnalités cartographiques, automatiser

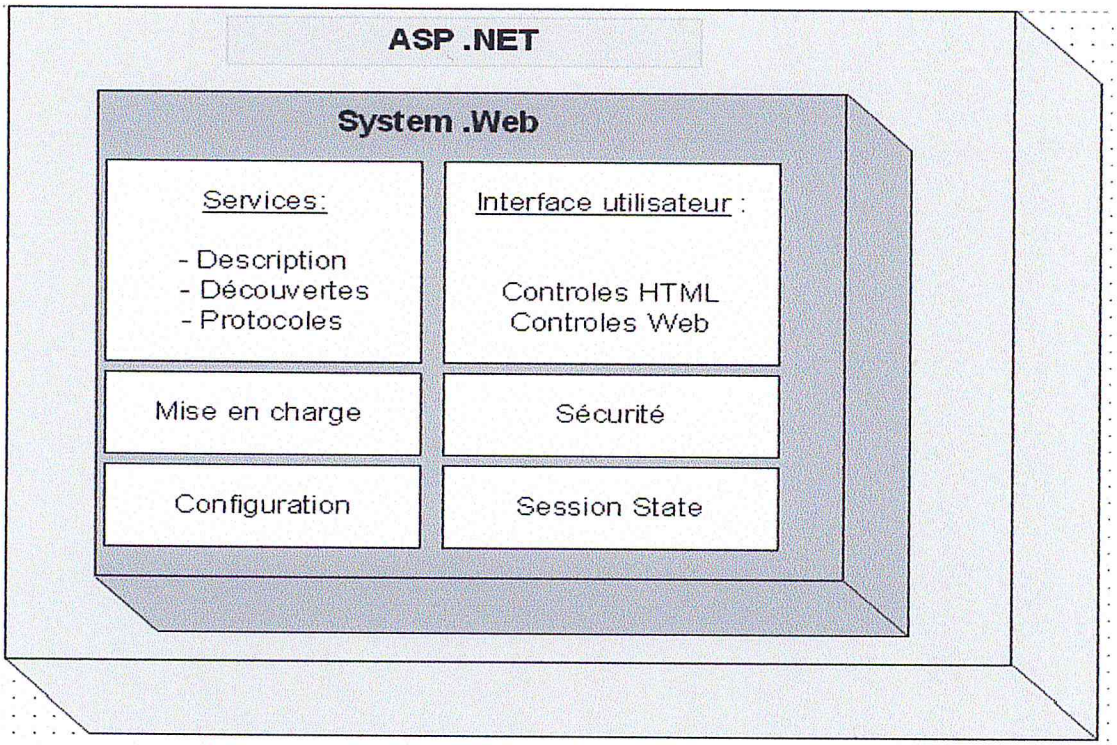


Figure A.7 : *Présentation des formulaire et services Web : ASP .NET*

Dans **System.Web,** certains services tels que la mise en charge, la sécurité ou la configuration sont partagés par les services Web et l'interface.

III.5- Interface utilisateur

La figure suivante explique comment le .NET gère l'interface Framework des applications Windows traditionnelles :

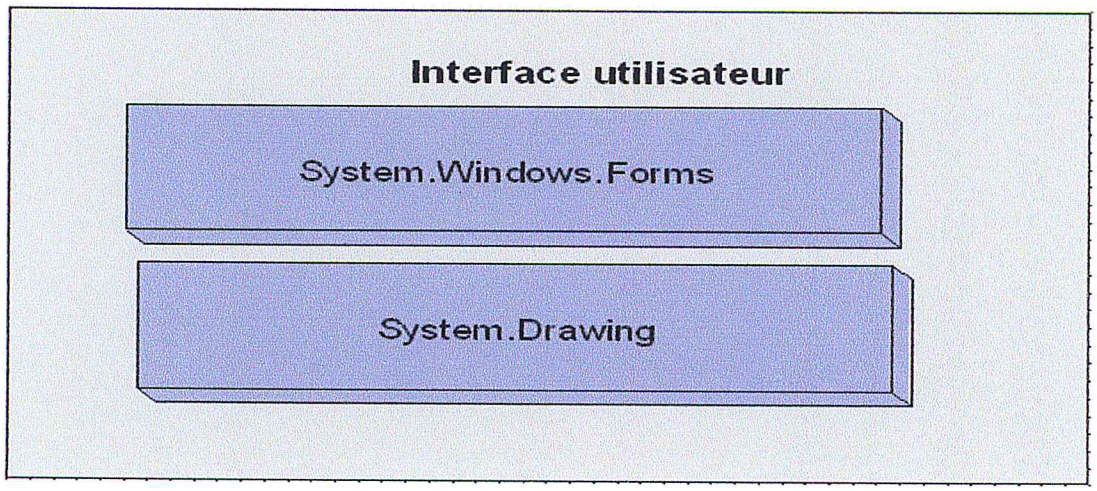


Figure A.8 *Interface utilisateur.*

des traitements répétitifs ou intégrer MapInfo dans d'autres applicatifs. MapBasic® contient des procédures permettant, en quelques lignes de code, d'intégrer la dimension géographique dans les applications en y exploitant des cartes et des fonctions cartographiques. Les programmes MapBasic® sont facilement intégrables dans des développements réalisés dans d'autres langages tels que Visual basic, Delphi, C++, PowerBuilder... La version 5 .5 de MapBasic fonctionne avec Map Info V 6.xx.

5. MapInfo MapX™ 4.5 est l'ActiveX cartographique des développeurs d'applications MapInfo.

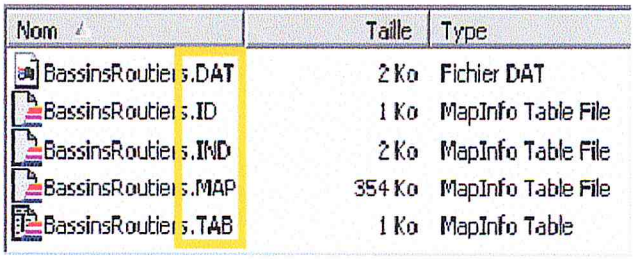
6. MapXtreme Java Edition pour Internet, est une solution cartographique " 100% Pure Java™ ". Portable sur toutes les plate-formes (Unix ou Windows NT), MapXtreme Java Edition pour Internet est une solution pour intégrer les applications cartographiques sur Internet/ Intranet.

3. Les données fournies avec le logiciel MapInfo Professional®

Construire un SIG, c'est compiler, assembler, croiser des données thématiques professionnelles avec des données génériques vecteurs ou raster. En standard, MapInfo Professional est livré avec quelques jeux de données. Mais celles ci sont à vocation purement pédagogique de manière à assurer la prise en main du logiciel par l'acquéreur.
Pour s'équiper des données qui conviennent le mieux aux applications particulières, il faut rechercher des données vers les producteurs institutionnels ou leurs distributeurs.

4. La structure des données au format MapInfo

MapInfo est un logiciel qui structure les informations en tables. Une table est un ensemble de fichiers qui sont manipulés ensemble par le logiciel.
Ainsi la fonctionnalité «Ouvrir une table » est traduite par un ensemble d'activités informatiques élémentaires qui vont ouvrir chacun des fichiers constituant la table, vérifier la cohérence de l'ensemble et afficher le contenu graphique de la table dans une fenêtre.
Ainsi, les informations communales gérées par MapInfo vont être constituées d'un certain nombre de fichier. C'est cet ensemble de fichier que l'on nomme «table des communes ».



Nom	Taille	Type
BassinsRoutiers.DAT	2 Ko	Fichier DAT
BassinsRoutiers.ID	1 Ko	MapInfo Table File
BassinsRoutiers.IND	2 Ko	MapInfo Table File
BassinsRoutiers.MAP	354 Ko	MapInfo Table File
BassinsRoutiers.TAB	1 Ko	MapInfo Table

*.MAP -> Regroupe les données géométriques décrivant les entités géographiques (Position, forme des objets, …)

*.DAT -> La base de données

*.ID -> Regroupe l'information permettant d'établir le lien entre les vecteurs et la base de données

(*.MAP) ⟷ (*.DAT)

*.IND -> Est l'index sur les données descriptives. Il existe seulement si au moins un champ de la table est indexé. Permet de faire des recherches plus rapidement.

*.TAB -> Fichier principal: Il relie l'ensemble des fichiers afin de les ouvrir dans MapInfo.

MapXtreme 2004

La solution pour le développement .NET.

MapXtreme 2004 permet de créer des applications intégrant une dimension cartographique aussi bien en environnement monoposte, client/serveur que web.

MapXtreme 2004 est un nouveau produit majeur dans l'offre d'outils de développement de MapInfo, le leader mondial des SIG sur PC.

MapXtreme 2004, c'est :

- Un produit identique pour le développement d'application mono-postes, client/serveur et web.
- Une grande facilité d'utilisation et de développement grâce à une intégration pertinente dans l'environnement de développement Visuel Studio.NET. Cette intégration apporte à l'outil Microsoft la possibilité d'afficher, traiter et modifier l'information cartographique.
- Une interopérabilité réelle par le respect de standards issus des domaines informatiques comme cartographique.
- Un outil avec de nombreuses fonctionnalités qui permet de répondre parfaitement à vos besoins.

1. Développement WEB

MapXtreme 2004 facilite le développement d'applications Web cartographiques en fournissant des modèles d'applications (templates), de multiples outils standards (zoom, recentrer,…), de nombreux exemples ainsi que le support du " cliquer-deposer " (drog and drop) dans votre environnement de développement. Tous les langages permettant la réalisation d'applications ASP.NET (comme VB.NET et C#) peuvent être utilisés avec MapXtreme 2004.

2. Développement monoposte

Tout comme le développement web, MapXtreme 2004 intègre des modèles d'applications (templates) qui configurent et chargent l'environnement de développement de manière automatique. Ceci inclus les outils cartographiques standard, des exemples de code et un environnement de développement supportant le " cliquer-deposer " (drog and drop).

En environnement monoposte, MapXtreme 2004 fonctionne avec tous les langages .NET comme VB.NET, C# et Visuel C++.NET.

Points forts de MapXtreme 2004

1. Facilité d'utilisation

- Intégration complète à l'environnement de développement Microsoft Visuel Studio.NET.
- Support du " cliquer-deposer " (drog and drop) dans votre environnement de développement.
- Modèles d'applications pré-définis et exemples d'applications pour permettre un développement rapide.
- Développement de vos applications avec le langage .Net de votre choix, pour maximiser l'exploitation de vos compétences informatiques.

2. Interopérabilité

- Support de standards informatiques comme Microsoft.NET, ADO.NET (et SQL-3 - via SpatialWare).
- Support de standards géomatiques issus de OpenLS et OpenGIS Consortium, comme GML, WMS et WFS.
- Accès aux données là où elles résident dans votre système d'information, grâce à l'utilisation de standards d'accès aux données comme ADO.NET, ODBC, Oracle Spatial, et Microsoft SQL Server.
- Intégration de la géolocalisation dans vos applications.

3. Riche en fonctionnalité

- Moteur de gestion de données très performant grâce à l'utilisation de plusieurs types de curseurs, d'interfaces standard pour l'accès aux données et l'utilisation du langage standard de requête.
- Serveur et client WMS (Web Map Server).
- Serveur et client WFS (Web Feature Server).
- Barre d'outils et boites de dialogue standards extensibles par le développeur.
- Symbologie riche : étiquettes curvilignes, grand choix de styles de remplissage et de symboles.
- Nouvelle logique d'organisation des cartes.

Les objets géographiques, les étiquettes et les analyses thématiques sont séparés afin de contrôler leur ordre et leur priorité de rafraîchissement.

PRE-REQUIS

Environnements supportés

Bases de données :

- MS Access
- Oracle 8i
- MS SQL Server 2000
- Oracle 9i

Systèmes d'exploitations :

- Windows 2000 SP4 Professional
- Windows 2000 SP4 Advanced Server
- Windows 2003 Server Standard Edition
- Windows 2003 Server Entreprise Edition
- Windows 2000 SP4 Server
- Windows XP Professional SP1
- Windows 2003 Server Web Edition

Environnements de développement :

- Microsoft .NET Framework 1.1
- Visual Studio .NET 2003
- Visual C#
- VB.NET

Configuration matérielle

Mémoire

192 Mo RAM minimum

Processeur (CPU) :

- Minimum : Pentium II – 450 Mhz
- Recommandé : Pentium III – 600 Mhz

La Suite graphique CorelDRAW® 12

La Suite graphique CorelDRAW® 12 innove en offrant des outils intelligents qui réduisent les étapes de la conception, simplifient le reciblage des graphismes, accélèrent la révision des projets, aident à mieux positionner textes et objets et optimisent les mises en page.

La suite graphique CorelDraw 12.0 est essentiellement composée de trois programmes – CorelDraw, Corel Photo-Paint et Corel RAVE – qui permettent de traiter tous les types de graphiques, dont les dessins vectorisés, les images au format bitmap et les animations web simples. Particulièrement puissant et étroitement intégré à éditeur d'images Corel Photo-Paint, le logiciel CorelDraw proprement dit peut gérer des illustrations complexes et des publications multipages. Un didacticiel sur CD et une interface claire facilitent grandement l'utilisation de la suite qui, en outre, se révèle particulièrement bon marché comparé à ses concurrents d'Adobe. Toutefois en raison du faible nombre d'améliorations apportées à Corel Photo-Paint et à Corel RAVE et du coût particulièrement élevé de la mise à jour, les utilisateurs réfléchiront peut-être deux fois avant de changer de version.

Au fil des versions, la suite CorelDraw continue de se mesurer à ses principaux concurrents Illustrator et Photoshop d'Adobe et réussit à maintenir sa position. Ce n'est pas étonnant car cette suite offre une interface brillamment conçue. La version 12 s'inscrit dans la même lignée: ses constituants CorelDraw (illustration), Corel Photo-Paint (retouche d'images numériques) et Corel RAVE (animation graphique) sont pourvus d'une interface très conviviale et totalement configurable. Espaces de travail et barres d'outils peuvent être créés et associés à des tâches spécifiques telles que la mise en page de brochures ou projet créatif, puis sauvegardés et réutilisés ultérieurement. D'où un gain de productivité significatif.

CorelDraw, Corel Photo-Paint et Corel RAVE offrent toute la gamme des outils graphiques, en quantité suffisante pour mener à bien tout type de projet. Dans cette nouvelle version, Corel en introduit un certain nombre, puissants, notamment pour le dessin et apporte un peu partout des petites modifications : meilleur contrôle des polices (accès facilité au jeu de caractères Unicode), amélioration du support des importations/exportations (exportation vers Microsoft Office et WordPerfect Office, filtres SVG, AutoCAD, Visio et HPGL). Sans oublier que la suite continue de permettre l'importation/exportation des formats de fichiers Adobe tels qu'EPS ou PSD.

1. Prise en charge des formats d'Adobe

L'intégration des trois applications principales est excellente. Ainsi, sous CorelDraw, pour retoucher une image bitmap contenue dans une brochure, il suffit de cliquer deux fois sur l'illustration pour lancer Corel Photo-Paint et, dans ce nouvel environnement, effectuer toutes les modifications nécessaires. Et lors du retour sous CorelDraw, l'image mise à jour est déjà là! De façon analogue, il est possible de faire glisser une image sous Corel RAVE pour ajouter des effets d'animation.

Les nouvelles possibilités offertes par la suite sont centrées sur la simplicité d'emploi. Ainsi, à l'intention des débutants, CorelDraw propose maintenant un outil de dessin assisté qui tente de transformer les ébauches tracées par l'utilisateur en formes géométriques courantes. Un rond très approximatif devient ainsi un cercle parfait. Cet outil fonctionne particulièrement bien avec les esquisses de cercles et de rectangles. Globalement, ce nouvel outil est pratique pour esquisser rapidement organigrammes et modèles de présentation.

Une autre nouveauté importante du logiciel CorelDraw concerne l'ancrage magnétique. Après l'activation de repères dynamiques, des lignes bleues temporaires s'affichent lors du passage de la souris sur certains points tels que le centre d'un objet. Ces guides se révèlent pratiques pour placer les éléments les uns par rapport aux autres.

Quant aux deux autres programmes de la suite, ils ne bénéficient que de peu de changements importants. Corel Photo-Paint propose un pinceau de retouche pour supprimer les défauts des photos (poussières, rayures et plis). Corel RAVE comporte une bibliothèque de symboles qui accueille les objets fréquemment utilisés (les symboles sont des objets préalablement définis auxquels il est possible de se référer plusieurs fois dans un même document, ce qui réduit d'autant la taille du fichier correspondant).

2. Caractéristiques et applications clés

2.1. Outils intelligents pour simplifier la conception graphique

Nouveauté! Outil Dessin assisté - La Suite graphique CorelDRAW 12 reconnaît les formes de base — cercles, triangles, flèches, parallélogrammes — et lisse, par exemple, les courbes

alors même que vous dessinez. Cette intelligence réduit les étapes de la conception graphique et enrichit vos possibilités créatrices.

Nouveauté! Croquis et esquisses éclairs – Esquisser rapidement une publicité ou une mise en page avec l'outil Dessin assisté de CorelDRAW 12. Cet outil est conçu pour corriger les imprécisions, positionner des objets et dessiner des formes de manière à écourter sensiblement le processus de révision.

Nouveauté! Repères dynamiques – Effectuer la mise en page des objets sans tâtonnement. Activés, les repères dynamiques affichent des repères temporaires qui, parce qu'ils sont liés aux points magnétiques des objets, accélèrent la conception et le reciblage des graphismes.

2.2. Améliorations qui favorisent des gains d'efficacité

Intégration complète et interface transparente – Passer aisément d'une application à l'autre — CorelDRAW 12, Corel PHOTO-PAINT® 12 et Corel R.A.V.E™ 3 — pour réaliser des mises en page et modifier de manière efficace objets vectoriels et images bitmap

Nouveauté! Outil Pinceau de retouche – Supprimer les imperfections de vos images en vous laissant guider par les indications en temps réel.

Amélioration! Personnalisation – Adapter les applications aux besoins des travaux en cours. La barre de menus, la barre des propriétés et la barre d'état sont autant d'éléments qui peuvent être personnalisés. A tout moment, il est aussi possible de faire glisser d'autres commandes vers l'espace de travail.

2.3. Flux de travail simplifié par la compatibilité accrue des fichiers

Compatibilité étendue – Échanger aisément vos fichiers, modifier les projets et partager vos créations en mettant à profit plus de 100 filtres d'importation et d'exportation dont SVG, AutoCAD®, HPGL, Adobe® Photoshop®, Adobe® Illustrator® et Adobe PDF.

Nouveauté! Exporter vers Microsoft® Office ou WordPerfect® Office – Publier vos fichiers aux formats supportés par Microsoft Office ou WordPerfect Office. Cette fonction comporte des options permettant d'incorporer aux documents des graphismes issus de la Suite graphique CorelDRAW 12.

Fichiers de production sur mesure faciles à déployer – Publier vos fichiers graphiques au format PDF pour les rendre compatibles avec les exigences des ateliers de composition. Les avertissements de vérification avancés vous aident à contrôler la sortie finale et à prévenir de fâcheux imprévus à l'étape de l'impression.

2.4. Suite graphique complète d'un rapport qualité-prix exceptionnel

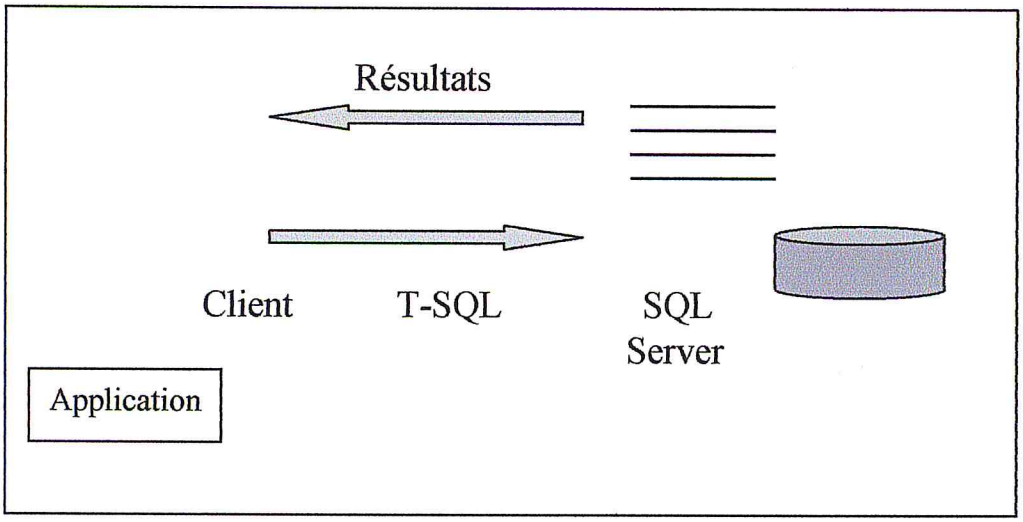
Solution complète – trois applications intégrées de premier choix et faciles à utiliser.
Utilitaires professionnels – Accéder sans détour à un ensemble d'outils professionnels dont CorelTRACE® 12, Corel CAPTURE™ 12, Microsoft® Visual Basic® pour Applications 6.3, le système de gestion des couleurs Kodak Digital Science™ et le lecteur QuickTime® 6.
Ressources pratiques – Simplifier votre flux de travail en accédant rapidement aux ressources numériques, aux modèles déclinés par étapes, aux didacticiels et à Mise à jour Corel, une fonction qui vous avise des mises à jour et les achemine directement à votre ordinateur.

SQL Server

I. Présentation de SQL Server

I.1. Définition

SQL Server est un système de gestion de bases de données relationnelles (SGBDR) Client-Serveur qui utilise ***Transact-SQL*** comme langage de requête entre un client et SQL Server.



Architecture Client-Serveur

I.2. Logiciels de SQL Server

SQL Server inclut plusieurs logiciels destinés à l'administration et la gestion du serveur, à la recherche de rubrique d'aide spécifique, à la conception et à la création de la base de données et à l'intégration des données.
SQL Server fournit un client administratif, **SQL Server entreprise manager.**

I.3. Assistants et outils graphiques

SQL Server fournit plusieurs Assistants et outils d'administration conçus pour l'utilisation de certaines fonctionnalités de SQL Server. On peut citer certains assistants :

- Analyseur de requêtes,
- Analyseur de performances SQL Server,
- Générateur de profils SQL Server,
- Utilitaire réseau du client,
- Gestionnaire de services SQL Server,
- Documentation en ligne.

II. Intégration SQL Server – Windows NT

Pourquoi le choix de SQL Server sur une plate forme Windows NT ?

II.1. Sécurité

SQL Server est intégré au système de sécurité de Windows NT. Cette intégration permet d'accéder à SQL Server et à Windows NT à partir d'un nom d'utilisateur et d'un mot de passe unique. SQL Server utilise également les fonctionnalités de cryptage de Windows NT pour la sécurité du réseau.

II.2. Prise en charge de plusieurs processeurs

SQL Server prend en charge les fonctionnalités de multi-traitement de Windows NT, SQL Server tire automatiquement parti de tout processeur ajouté à l'ordinateur serveur.

II.3. Observateur d'événement

SQL Server décrit des messages dans les journaux d'évènements système, sécurité et applications de Windows NT. Offrant ainsi un mécanisme cohérent pour l'affichage et le suivi des problèmes.

II.4. Services Windows NT

SQL Server est exécuté en tant que service sous Windows NT, ce qui nous permet de démarrer et d'arrêter SQL Server à distance.

II.5. Analyseur de performances Windows NT

SQL Server transmet des informations de performance Windows NT, ce qui permet de surveiller les performances système de SQL Server.

III. Architecture de SQL Server

Toutes les caractéristiques et les composants de SQL Server collaborent pour accroître son efficacité en tant que base de données relationnelle, client/serveur, basée sur le langage SQL . Les diverses architectures de SQL Server :

_Architecture Client/Serveur,

_Architecture de base de données,

_Architecture d'administration,

_Architecture de développement d'application,

_Architecture de réplication.

VI. Fichiers composants une base SQL Server

Les bases de données SQL Server possèdent trois types de fichiers :

VI.1. Fichiers de données primaires

Ces fichiers sont le point le départ de la base de données et pointent sur les autres fichiers de la base de données. Chaque base de données comprend un fichier de données primaire. L'extension de fichier recommandée est **.mdf**.

VI.2. Fichiers de données secondaires

Ces fichiers comprennent tous les fichiers de données autres que le fichier de données primaires. Certaines bases de données possèdent plusieurs fichiers de données secondaires. L'extension recommandée est **.ndf**.

VI.3. Fichiers journaux

Ces fichiers contiennent toutes les informations de suivi nécessaires à la reprise de la base de données. L'extension recommandée est **.ldf**.

V. Les bases de données système

Le système SQL Server possède quatre bases de données système :

BD master : elle enregistre l'intégralité des informations systèmes relative à un système SQL Server, elle enregistre aussi l'existence de toutes les bases de données.

BD tempdb : elle contient toutes les tables et les procédures stockées temporaires. Toutes les tables temporaires de tous les utilisateurs connectés au système y sont stockées. Elle est recréée à chaque fois que le SQL Server est lancé. Les tables sont automatiquement supprimées à la déconnexion.

BD model : la Bd model fait office de modèle pour toutes les bases de données créées sur un système lors de l'émission d'une instruction CREATE DATABASE.

BD msdb : elle permet à SQL Server Agent de planifier les alertes et les travaux et d'enregistrer les opérateurs.

IV. Les instructions SQL

IV.1. Les instructions DCL

Les instructions DCL permettent de modifier les autorisations associées à un utilisateur ou un rôle de base de données.

GRANT : crée une entrée dans le système de sécurité qui permet à un utilisateur de travailler.
DENY : crée une entrée dans le système de sécurité qui refuse des autorisations d'un compte de sécurité.
REVOKE : supprime une autorisation précédemment accordée ou refusée.

IV.2. Les instructions DDL

Les instructions DDL définissent la base de données par la création de base de données, de table et de type de données définis par l'utilisateur. On peut aussi utiliser des instructions DDL pour gérer les objets de la base de données.
Les instructions suivantes sont des instructions DDL :
CREATE nom_objet
ALTER nom_objet
DROP nom_objet

IV.3. Les instructions DML

Les instructions DML utilisent les données de base de données. Les instructions DML permettent de modifier des données ou d'extraire des informations.
Les instructions DML sont les suivantes :
SELECT, INSERT, UPDATE, DELETE.
Par défaut, seuls les membres des rôles sysadmin, db_owner peuvent exécuter les instructions DML.

IIV. Caractéristiques de SQL Server

Parmi ses principales caractéristiques :

- La gestion des grands volumes de données.
- Traitement simultané des demandes de données venant des logiciels clients et ne retourne que les données demandées.

- Verrouillage sélectif des données pour permettre aux utilisateurs de travailler sur différentes parties sur un même ensemble de données en même temps.

IIIV. Les commandes SQL

En SQL, les commandes sont dites de format libre. Cela signifie qu'il n'existe aucune règle indiquant qu'un mot donné doit commencer à une position particulière de la ligne. Cette manière d'écriture est plus lisible.

Quelques commandes SQL :

CREATE Table : permet de créer une table,
DROP Table : pour supprimer une table,
INSERT : en ajoutant des lignes aux colonnes, on encadre les valeurs par des apostrophes,
SELECT : permet d'afficher les données sélectionnées,
UPDATE : pour modifier la valeur d'une donnée,
DELETE : pour supprimer un enregistrement.

Des conditions sont associées à ces commandes grâce à la clause **WHERE**, elle est utilisée pour retrouver les enregistrements qui répondent à une certaine condition. Cette condition est appelée simple. En connectant plusieurs conditions simples par l'utilisation des opérateurs : **AND**, **OR** et **NOT**, on aura les conditions combinées.
La clause **IN** est une manière concise de formuler certaines conditions.

L'opérateur **BETWEEN** n'est pas une fonctionnalité essentielle de SQL, mais il rend toutefois certaines commandes SELECT plus simples.
Dans la plupart des cas, les conditions nécessitent une concordance exacte. Pour cela, l'opérateur **LIKE** avec un caractère joker est utilisé.

L'ordre des lignes dans une table est sans importance dans un SGBD. D'un point de vue pratique, lors de l'interrogation d'une base de données relationnelle, le résultat s'affiche sans ordre prédéfini. Les lignes peuvent s'afficher dans l'ordre choisis spécifié à l'aide de la clause **ORDER BY**.

SQL dispose de fonctions pour calculer :
la somme : en utilisant la fonction **SUM** ,
la moyenne : grâce à la fonction **AVG** ,
ainsi que pour compter, on utilise **COUNT** ,
et pour trouver la valeur du maximum et du minimum, en introduisant les fonctions **MAX** et **MIN** respectivement.
L'opérateur **DISTINCT** n'est pas une fonction, mais il peut néanmoins se révéler utile dans certaines situations, couplé avec la fonction **COUNT**.

La clause **GROUP BY** permet de grouper des données dans un ordre particulier puis de calculer des statistiques si nécessaire.
La clause **HAVING** est utilisée pour les groupes, en effet, elle fait pour les groupes ce que la commande WHERE fait pour les lignes.

En SQL, on réalise la jointure entre tables en incluant une condition dans la clause WHERE de façon à s'assurer que les colonnes en concordance contiennent des valeurs égales. Pour réaliser la jointure, on utilise des sous requêtes avec **IN** et **EXISTS**.

XI. Gestion de la sécurité par l'administrateur

La sécurité consiste en la prévention des accès non autorisés à une base de donnée.
Dans une organisation donnée, c'est l'administrateur de base de données qui détermine les types d'accès à la base dont les divers utilisateurs ont besoin. Certains utilisateurs peuvent consulter et mettre à jour les données, d'autres consulter des données dans la base mais non les modifier, d'autres encore n'avoir accès qu'à une partie de la base.

Une fois déterminées, ces règles d'accès sont mises en œuvre par le mécanisme de sécurité fourni par le SGBD. Dans les systèmes SQL, les vues offrent une certaine sécurité, puisque, si un utilisateur accède à la base à travers l'une d'entre elles, il ne peut accéder à des données qui ne font pas partie de cette vue. Le mécanisme principal pour la sécurité est toutefois la commande GRANT. Elle s'appuie sur le fait que l'administrateur de base de données peut accorder différents types de privilèges, ou droits, aux utilisateurs et les révoquer ultérieurement, si nécessaire. Ces privilèges incluent la capacité de consulter les lignes d'une table, d'insérer de nouvelles lignes, de mettre à jour des lignes existantes, etc…
On peut accorder et révoquer ces privilèges à l'aide des commandes GRANT et REVOKE.

Internet Information Serveur (I.I.S.)

L'un des problèmes de base des services W.W.W. avec la programmation de documents HTML est l'aspect statique des pages qui ne permettent que la consultation d'informations.

La professionnalisation d'Internet et du World Wide Web (WWW) a permis la création de serveur WWW assurant l'exécution de programme, et l'accès aux données. Les documents visualisés par les navigateurs Web sont ainsi devenus dynamiques.

Cette présentation repose sur le système propriétaire MicroSoft, avec l'utilisation de Windows NT server 4.0 qui offre un serveur W.W.W. à l'installation par l'intermédiaire d'Internet Information Serveur (version 3 après correctif du service pack 3 I.I.S.3), qui permettra l'exécution de programmes ou de gérer l'accès à des bases de données.
En préalable on considère l'installation de Windows NT et donc du serveur WWW complète, ainsi que le serveur WWW démarré.

Le serveur World Wide Web I.I.S. fournit avec Windows NT 4.0 est à la base du développement et la gestion de sites Internet ou Intranet, c'est à dire qu'il va permettre la publication de documents au format ASPX, la gestion de formulaires interactifs avec des applications, des bases de données, ..., ainsi que la gestion de transferts de fichiers, ou encore la publication d'archives d'informations.

Les composants de IIS :

Internet Information Serveur inclut les composants suivants :

- Les services Internet :

- **WWW** (World Wide Web) est le service qui permet la gestion des documents HTML, mais aussi certains accès aux bases de données.
- **GOPHER** est un service plus ancien et aujourd'hui très peu utilisé qui permet d'organiser un système d'information où les fichiers sont classés en fonction d'une arborescence (sur le modèle des répertoires sous DOS).
- **FTP** (File Transfert Protocol) est le service qui va permettre les transferts de fichiers.
- le Gestionnaire des services Internet, qui permet d'administrer les services Internet,

- Le connecteur de base de données Internet (IDC, *Internet Database Connector*), qui sert à envoyer des demandes aux bases de données,

- La gestion de l'Active Server Page (A.S.P.) de la technologie DCOM de MicroSoft, qui va servir à une programmation dynamique, et un accès aux bases de données par l'utilisation du modèle Active Data Object (A.D.O.),

- Le Gestionnaire de clés, qui sert à installer les clés de sécurité SSL (*Secure Sockets Layer*).