

MINISTRE DES ENSEIGNEMENTS SECONDAIRE,
SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
(M.E.S.S.R.S)

Burkina Faso

UNIVERSITE POLYTECHNIQUE
DE BOBO – DIOULASSO
(U.P.B)

Unité – Progrès – Justice

ECOLE SUPERIEURE D'INFORMATIQUE (E.S.I)
01 BP 1091 TEL: (226) 97-27-64
BOBO-DIOULASSO

THEME :

**MISE EN ŒUVRE D'UN SYSTEME DE
GESTION DES ENSEIGNEMENTS.**

MEMOIRE DE FIN D'ETUDE

Présenté et soutenu publiquement le 15 février 2001

Pour l'obtention du

Diplôme d'ingénieur de conception en informatique

Par

Yacouba OUATTARA

Composition du jury

Président : Bétaboalé NAON

Rapporteur : Oscar MANSO

Directeur de mémoire : Mesmin DANDJINO

SOMMAIRE

REMERCIEMENT 3

INTRODUCTION 4

I. LA MOTIVATION DE CETTE ETUDE 5

II. ATTENTE DES UTILISATEURS..... 5

 II.1. LA SITUATION ACTUELLE..... 5

 II.2. LA DEMANDE DE L'UTILISATEUR..... 6

III. LE MODELE DU SYSTEME 6

 III.1. LA DEMARCHE UTILISEE 7

 III.2. L'ANALYSE CONCEPTUELLE DU SYSTEME 10

 III.3. LES PRINCIPAUX TRAITEMENTS..... 13

 III.4. LE DEPLOIEMENT DE L'APPLICATION SUR LE RESEAU 16

IV. L'ARCHITECTURE DU SYSTEME..... 19

 IV.1. ARCHITECTURE GLOBALE DU SYSTEME..... 19

 IV.2. ARCHITECTURE DU SYSTEME DE GESTION DES ENSEIGNEMENTS 23

 IV.3. LE MOTEUR D'INFERENCE..... 24

V. CHOIX DES OUTILS..... 29

 V.1. CHOIX DE LA BASE DE DONNEES 29

 V.2. CHOIX DU LANGAGE 35

 V.3. CHOIX DU RESEAU (ETHERNET OU TOKEN RING)..... 37

VI. DETERMINATION DU COUT DU SYSTEME..... 42

 VI.1. ESTIMATION DU COUT DE DEVELOPPEMENT 43

 VI.2. ESTIMATION DU COUT DU MATERIEL..... 45

VII. REALISATION 46

CONCLUSION 51

BIBLIOGRAPHIE 52

SCHEMAS UTILISES 53

GLOSSAIRE..... 54

REMERCIEMENT

Ce document est le fruit de tant d'années d'étude. Il me donne l'occasion de remercier tous les enseignants de l'ESI ayant participé d'une manière ou d'une autre à ma formation.

Mes remerciements vont particulièrement à mon directeur de mémoire Mr Mesmin DANDJINOUI qui a pu aménager son temps pour me guider dans la réalisation de ce projet.

Mes remerciements s'adressent aussi au centre de calcul et plus précisément à Madame KABA et Mr Abdoul DIALLO qui m'ont permis d'avoir de quoi travailler.

Ces mêmes remerciements vont également à l'endroit des parents, amis et camarades étudiants qui n'ont cessé de me soutenir dans toutes mes études.

Que la paix du seigneur soit avec eux !

INTRODUCTION

L'Ecole Supérieure d'Informatique (E.S.I) de l'Université Polytechnique de Bobo-Dioulasso (U.P.B) comporte deux cycles :

- un Cycle d'Ingénieur de Travaux Informatiques (C.I.T.I) ouvert en 1990 avec une option analyste-programmeur et une option de maintenance effective depuis la rentrée 2000 ;
- un Cycle des Ingénieurs de Conception en Informatique (C.I.C.I) ouvert en 1998.

Dans le Cycle d'Ingénieur de Conception en Informatique, les étudiants en 3^{ème} année d'étude sont soumis à un stage d'étude de six mois dont le thème concerne généralement l'ingénierie de logiciels ou de réseaux informatiques.

C'est ainsi qu'au cours de l'année académique 1999-2000, nous nous sommes attelés à la « réalisation d'un système de gestion des enseignements » pour l'U.P.B qui traite en général de l'établissement des plannings annuel, mensuel ou hebdomadaire permettant la réalisation des enseignements à l'université. C'est une tâche très minutieuse qui demande un dévouement de la part des directeurs des études des facultés et écoles.

Ce document est notre mémoire de fin d'étude qui présente le fruit de nos réflexions et réalisations. En partant de ce qui motive cette étude, on a spécifié les attentes des utilisateurs et enfin fourni les résultats des travaux effectués depuis les étapes de la conception jusqu' à la validation de notre système.

I. LA MOTIVATION DE CETTE ETUDE

La gestion des enseignements peut être décrite en deux étapes :

- Une première étape portant sur l'affectation des matières et niveaux d'enseignement aux enseignants au début de l'année scolaire ou universitaire ;
- une deuxième étape qui comprend plusieurs phases :
 - une phase qui consiste à faire des prévisions pour le démarrage ou la fin d'un cours, ceci en perspective de la fin de l'année universitaire ou scolaire ;
 - une autre phase qui consiste à établir un programme pour l'exécution des cours dans les classes et salles. Cette partie est faite dans les écoles secondaires au début de l'année (elle peut subir de légères modifications au début de l'année) et est respectée jusqu'à la fin de l'année scolaire. Généralement, les salles sont en nombre suffisant et sont affectées aux classes pour toute l'année. Ce qui amoindrit la complexité de la programmation car il y a moins de paramètres à prendre en compte. Mais dans certains établissements et facultés de l'université, la programmation n'est pas annuelle mais hebdomadaire et doit tenir compte des paramètres tels que la disponibilité des enseignants, la disponibilité des élèves ou étudiants, des contraintes de salles, des contraintes de temps, du pourcentage des cours effectués. Il faut comprendre par contraintes de salles la disponibilité des salles d'une part et d'autres parts le type de salle ((salle de cours), de TP,...) et le nombre d'étudiants qu'elles peuvent recevoir.

C'est cette dernière phase qui pousse à la réalisation d'un système de production d'emploi du temps. Nous situant dans le domaine d'ingénierie du logiciel, on pourrait concevoir un système capable de reproduire le raisonnement des directeurs des études dans leurs activités de réalisation d'emploi du temps.

La réalisation d'un tel outil permettra non seulement de soulager les directeurs des études de l'U.P.B., mais aussi d'autres responsables de la planification des enseignements dans des établissements du secondaire.

II. ATTENTE DES UTILISATEURS

II.1. LA SITUATION ACTUELLE

L'U.P.B se compose de trois établissements. Aucun des établissements ne possède actuellement un outil spécifique de programmation des enseignements à part les outils de bureautique. La programmation des enseignements est effectuée par chaque directeur des études. Elle consiste à :

- établir les fiches programmatiques, c'est - à - dire à affecter aux enseignants les matières et les niveaux à enseigner au cours d'une année académique ;
- réaliser l'emploi du temps hebdomadaire grâce à la connaissance :
 - des fiches programmatiques ;
 - des matières à enseigner (programmes d'enseignements dans les différentes options, des différents niveaux des différents cycles) ;

- de l'ordre de précedence entre les matieres ;
 - de la disponibilite des salles ;
 - des contraintes des salles ;
 - des contraintes des enseignants ;
 - du calendrier officiel de l'annee universitaire ;
 - des congés et des fetes legales du pays.
- etabli les états périodiques de réalisation des cours : Les états périodiques sont effectués mensuellement ou à la fin d'une mission d'enseignement. Ils consistent d'une part à calculer le solde des heures de vacation en fonction du coût horaire qui dépend du grade de l'enseignant. D'autre part, ils consistent à faire les états de situation des cours en examinant les cahiers de texte pour prévoir la fin ou le démarrage d'un cours ;
 - etabli la fiche financière des enseignants vacataires, missionnaires ou permanents après une mission ou à la fin du mois.

II.2. LA DEMANDE DE L'UTILISATEUR

Tous les établissements de l'U.P.B souhaitent disposer d'un logiciel de programmation des enseignements pour alléger le travail des responsables qui en sont chargés (directeur des études, chef de section, chef de département,...).

Ce logiciel devra permettre de :

- ✳ saisir la fiche programmatique d'un enseignant ;
- proposer un emploi du temps hebdomadaire ;
- etabli les états de situations périodiques ;
- consulter des emplois du temps des niveaux d'enseignement ;
- ✳ consulter des matieres enseignées dans les niveaux d'enseignement ;
- ✳ saisir et de mettre à jour les contraintes d'un enseignant ;
- saisir et de mettre à jour les contraintes de salles ;
- mettre à jour les emplois du temps.

L'ensemble de ces fonctionnalités devra être accessible localement ou via l'Internet.

Il apparaît que ce logiciel devra faire des inférences à partir d'un certain nombre d'informations contextuelles pour proposer un emploi du temps hebdomadaire. Il devra analyser et raisonner sur les contraintes des enseignants, sur la disponibilité des salles et classes, et sur l'ordre de précedence qui pourrait exister entre les matieres.

III. LE MODELE DU SYSTEME

On cherche à donner une description complète du système d'information qui soit indépendante de la notion d'outil informatique. C'est une vue du système à très haut niveau, dans laquelle les fonctions utilisateurs majeures seront identifiées et leurs relations documentées. Pour cela on fera usage des diagrammes qui sont en général compréhensibles même par des utilisateurs n'ayant pas d'expériences techniques en informatique.

III.1. LA DEMARCHE UTILISEE

Trois approches ont été utilisées pour cerner tous les aspects du futur système :

- l'approche par les données ;
- l'approche fonctionnelle ;
- et l'usage de techniques de l'intelligence artificielle.

III.1.1. L'APPROCHE PAR LES DONNEES

C'est une méthode systémique d'analyse et de conception des systèmes d'information. Elle privilégie le système d'information en terme d'entité et de relation pour aboutir à un modèle cohérent de données. Elle nous permettra d'avoir le Modèle Conceptuel de Données. Le modèle conceptuel de données sert à formaliser la description des informations mémorisées par le système d'information et qui présentent un intérêt pour le domaine d'étude. Il permet d'exprimer également les contraintes d'intégrité ainsi que les liens sémantiques qui doivent être maintenus sur les données qui sont décrits par les règles de gestion. Les concepts de base pour cette modélisation sont les suivants :

- le concept de **propriété** : il caractérise l'information élémentaire qui doit être mémorisée dans le système d'information ;
- le concept d'**entité** : ce concept permet de conceptualiser dans le système d'information des objets matériels ou immatériels qui ont un intérêt pour le domaine considéré. Une entité définit une classe d'objets caractérisée par un ensemble de propriétés élémentaires choisies de manière à représenter une information particulière par rapport au domaine considéré. Une occurrence d'entité doit être constituée d'une et une seule occurrence de chacune des propriétés de l'entité. En d'autres termes, une propriété ne peut pas être omise, ni répétée dans une occurrence d'entité ;
- L'**identifiant** est une propriété particulière permettant de déterminer de façon unique une occurrence d'entité ;
- le concept de **relation** : il permet de conceptualiser les liens sémantiques qui peuvent exister entre les occurrences d'entités et qui ont un intérêt pour le domaine. Une relation est définie entre les entités ;
- Le concept de **cardinalité** : ce concept permet d'associer à chaque lien entre une entité et une relation une cardinalité minimale et une cardinalité maximale. La cardinalité minimale indique le nombre minimum de fois que toute occurrence d'entité peut participer à des occurrences de relations. La cardinalité maximale indique le nombre maximal de fois que toute occurrence d'entité peut participer à des occurrences de relation.

Nous pouvons résumer ces concepts par le schéma de la figure 1.

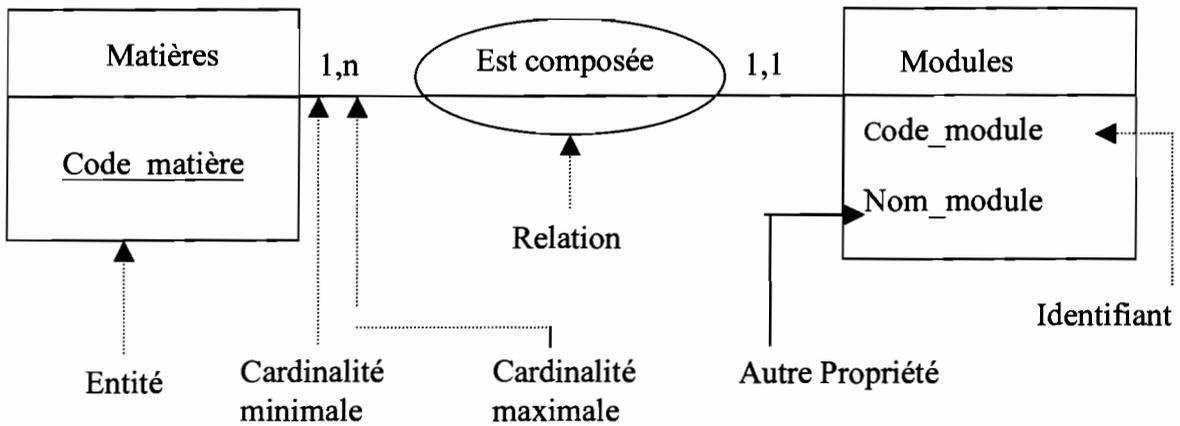


Figure 1 : illustration des concepts du modèle entité-association.

III.1.2. APPORT DE METHODES FONCTIONNELLES

Basées sur les diagrammes de flux, les méthodes fonctionnelles décrivent les processus de transformation au sein de l'application, c'est-à-dire les fonctions. Ces processus correspondent à une formalisation de scénarios dans lesquels on retrouve des acteurs, d'autres processus, des réceptifs et des flux de données.

Ces quatre concepts fondamentaux sont définis comme suit :

- Un **acteur** représente un producteur ou un consommateur de flux d'informations. L'activité de chaque acteur peut être décrite par un diagramme de flux de données. Un acteur peut aussi être assimilé à une fonction. On ne s'intéresse pas à sa description détaillée. Un acteur est représenté graphiquement par un rectangle.
- Un **flux d'information** représente un échange de données entre deux acteurs. Les flux en entrée d'un acteur peuvent être interprétés comme des événements auxquels cet acteur doit réagir, et les flux en sortie peuvent être interprétés comme des résultats de son activité. Un flux sera représenté par une flèche.
- Un **processus** est une opération de transformation des flux de données. Il sera représenté par une ellipse.
- Un **réceptif de données** est un moyen d'archivage des flux de données lorsqu'ils ne sont pas immédiatement consommés. Mais c'est aussi un réservoir d'information indépendant des différents acteurs. Il sera représenté par un cylindre.

Nous pouvons illustrer ces concepts par le schéma de la figure 2 ci-dessous :

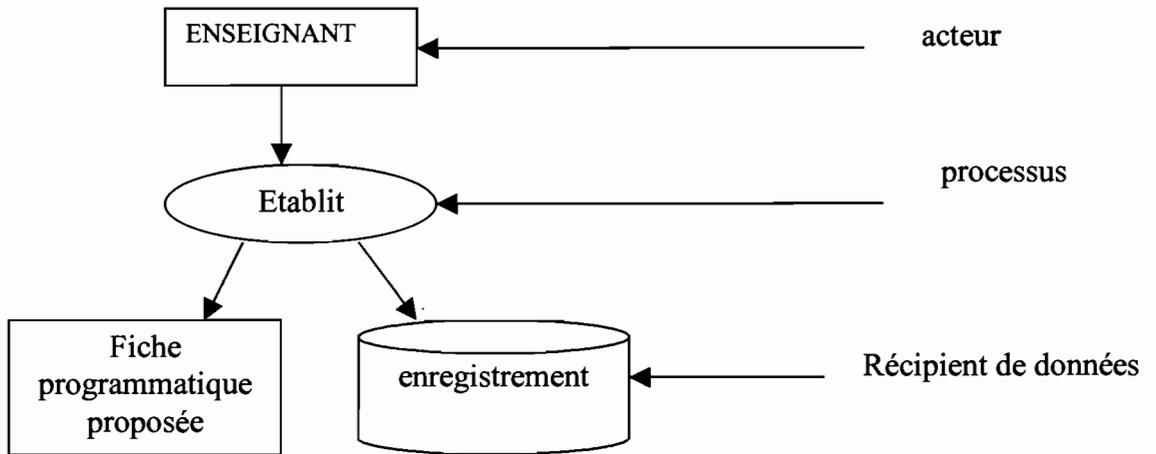


Figure 2 : illustration des concepts de méthodes fonctionnelles

III.1.3. APPORT DES TECHNIQUES DE L'INTELLIGENCE ARTIFICIELLE

Aujourd'hui, l'ordinateur est capable de battre en jeu d'échec des grands maîtres de ce jeu. D'autres arrivent à donner des conseils médicaux à des patients comme le feraient de vrais docteurs en médecine. Ces machines qui possèdent en elles - mêmes une capacité d'analyser, de raisonner comme un être humain, sont qualifiées de systèmes intelligents. Les systèmes experts, la conception assistée par l'ordinateur (C.A.O) sont des illustrations de systèmes intelligents.

Avec les techniques de l'intelligence artificielle, on essaiera de construire le moteur d'inférence pour la génération des emplois du temps hebdomadaire. La stratégie utilisée sera une stratégie mixte de chaînage avant et de chaînage arrière.

Un système qui part d'une base de faits qui ne contient que des assertions vraies, puis qui utilise des règles pour modifier cette base de faits, mais en ne laissant dans la base, à chaque étape, que des faits vrais pour atteindre un but est un système à chaînage avant. Le chaînage avant peut être défini comme un raisonnement guidé par des données pour aboutir à une solution. On peut inverser le raisonnement et partir du ou des buts à démontrer. C'est le chaînage arrière qui consiste à partir d'une solution de trouver les règles et données qui aboutissent à la solution. Nous utiliserons la technique du chaînage avant pour proposer un emploi du temps hebdomadaire et le chaînage arrière pour vérifier les modifications apportées à la proposition.

III.2. L'ANALYSE CONCEPTUELLE DU SYSTEME

III.2.1. LE MODELE CONCEPTUEL DE DONNEES(M.C.D)

Comme on aurait pu le prévoir au cours du modèle se trouve l'entité ENSEIGNANT. Les autres entités présentées sont sur la figure 3.

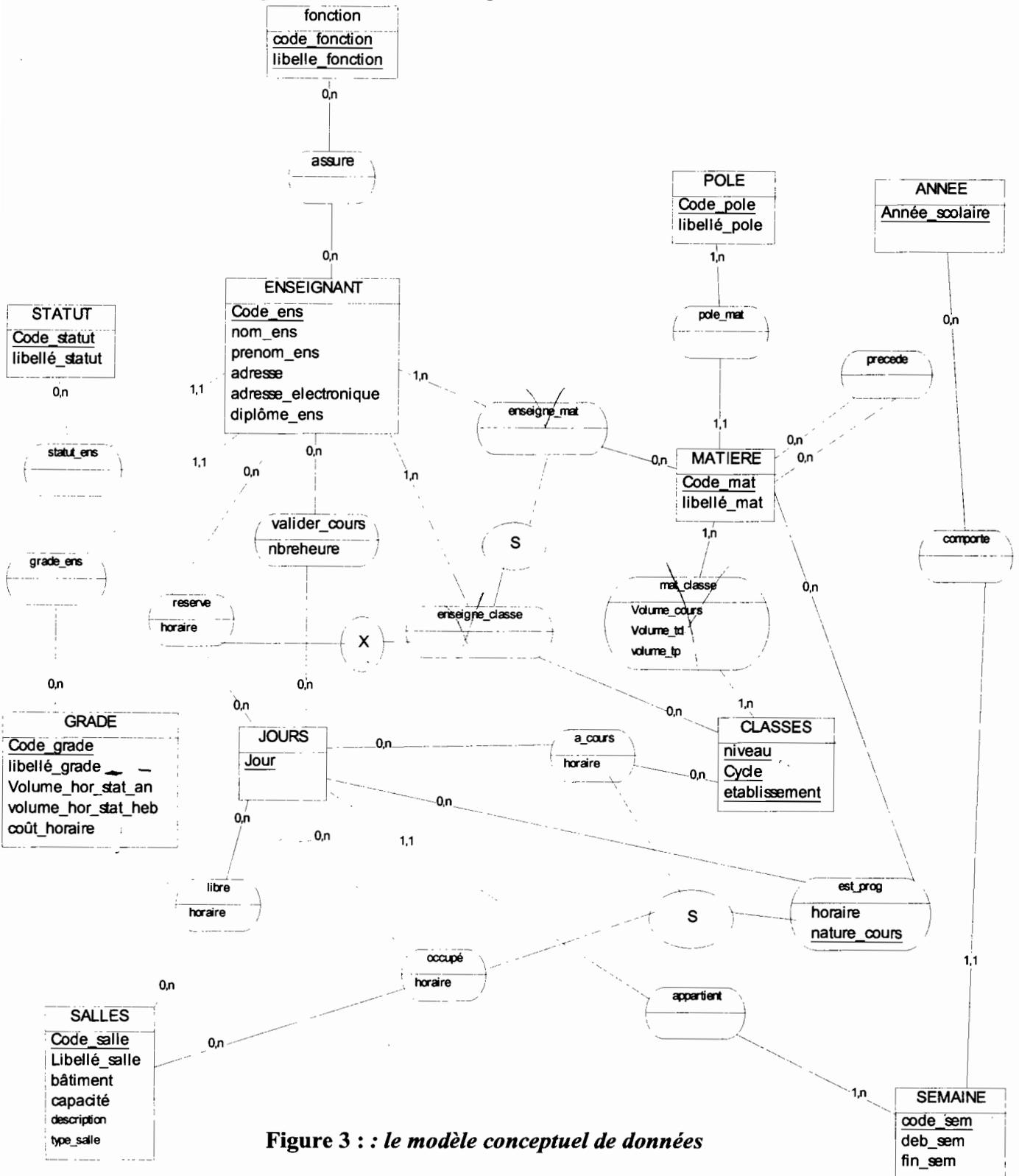


Figure 3 : : le modèle conceptuel de données

III.2.2. LE DICTIONNAIRE DE DONNEES

L'ensemble des données modélisés dans notre système figure dans le tableau 1.

PROPRIETE	TYPE	DESCRIPTION
Code_ens	AN	Le code de l'enseignant
Nom_ens	A	Le nom de l'enseignant
Prenom_ens	A	Le prénom de l'enseignant
Diplôme	A	Le diplôme de l'enseignant
Code_statut	AN	Le code du statut d'un enseignant
Libelle_statut	A	Le libellé de l'enseignant
Code_fonction	AN	Le code de la fonction
Libellé_fonction	AN	Le libellé de la fonction
Code_grade	AN	Le code de grade des enseignants
Libelle_grade	A	Le libellé d'un grade d'enseignant
Volume_hor_stat_an	N	Le volume horaire statutaire annuel
Volume_hor_stat_heb	N	Le volume horaire statutaire hebdomadaire
Coût_horaire	N	Le coût horaire pour une grade
Code_statut	AN	Le code du statut d'un enseignant
Libelle_statut	A	Le libellé du statut
Etablissement	A	L'école
Classe	AN	La classe
Cycle	AN	Le cycle
Code_mat	AN	Le code de la matière
Libelle_mat	A	Le libellé de la matière
Code_pole	AN	Le code d'un pôle d'enseignement
Libelle_pole	AN	Le libellé du pôle d'enseignement
Volume_cours	N	Le nombre d'heures de cours théorique
Volume_TD	N	Le nombre d'heures de travaux dirigés
Volume_TP	N	Le nombre d'heures de cours pratiques
Code_salle	AN	Le code d'une salle
Libelle_salle	AN	Le libellé de la salle
Bâtiment	AN	Le nom du bâtiment
Capacité	N	La taille du bâtiment
Type_salle	AN	Le type de salle
Jour	D	La date
Code_sem	N	Le code de la semaine
Deb_sem	D	La date de début de la semaine
Fin_sem	D	La date de fin de semaine
Année	AN	L'année universitaire
Horaire	AN	L'horaire de cours
Adresse_ens	AN	L'adresse de l'enseignant
Adresse_electronique_ens	AN	L'adresse électronique de l'enseignant

Tableau 1 : récapitulatif des informations du M.C.D.

III.2.3. QUELQUES REGLES DE GESTION

Les règles de gestion permettent de respecter les cardinalités qui existent entre les occurrences d'entité et de relation.

- Règle 1 : Une matière appartient à un pôle d'enseignement.
- Règle 2 : Une matière peut avoir des matières de précédence.
- Règle 3 : Un enseignant a une statut.
- Règle 4 : Un enseignant a un grade.
- Règle 5 : Un enseignant enseigne un ou plusieurs matières.
- Règle 6 : Un cours est programmé dans une salle.
- Règle 7 : Plusieurs matières peuvent être programmées dans des salles et classes différentes au même horaire et à la même date.
- Règle 8 : Des salles peuvent être libres à la même heure et à la même date.
- Règle 9 : Une matière est au moins enseignée dans une classe.
- Règle 10 : Une seule matière peut être enseignée par plusieurs enseignants durant une année.

III.3. LES PRINCIPAUX TRAITEMENTS

Trois principaux processus ont été identifiés : l'établissement de la fiche programmatique, l'établissement de la fiche financière et la proposition d'emploi du temps hebdomadaire.

III.3.1. LE PROCESSUS D'ETABLISSEMENT DE LA FICHE PROGRAMMATIQUE

Ce processus est généralement effectué au début d'une année académique et permet d'affecter aux enseignants les matières et les niveaux enseignés.

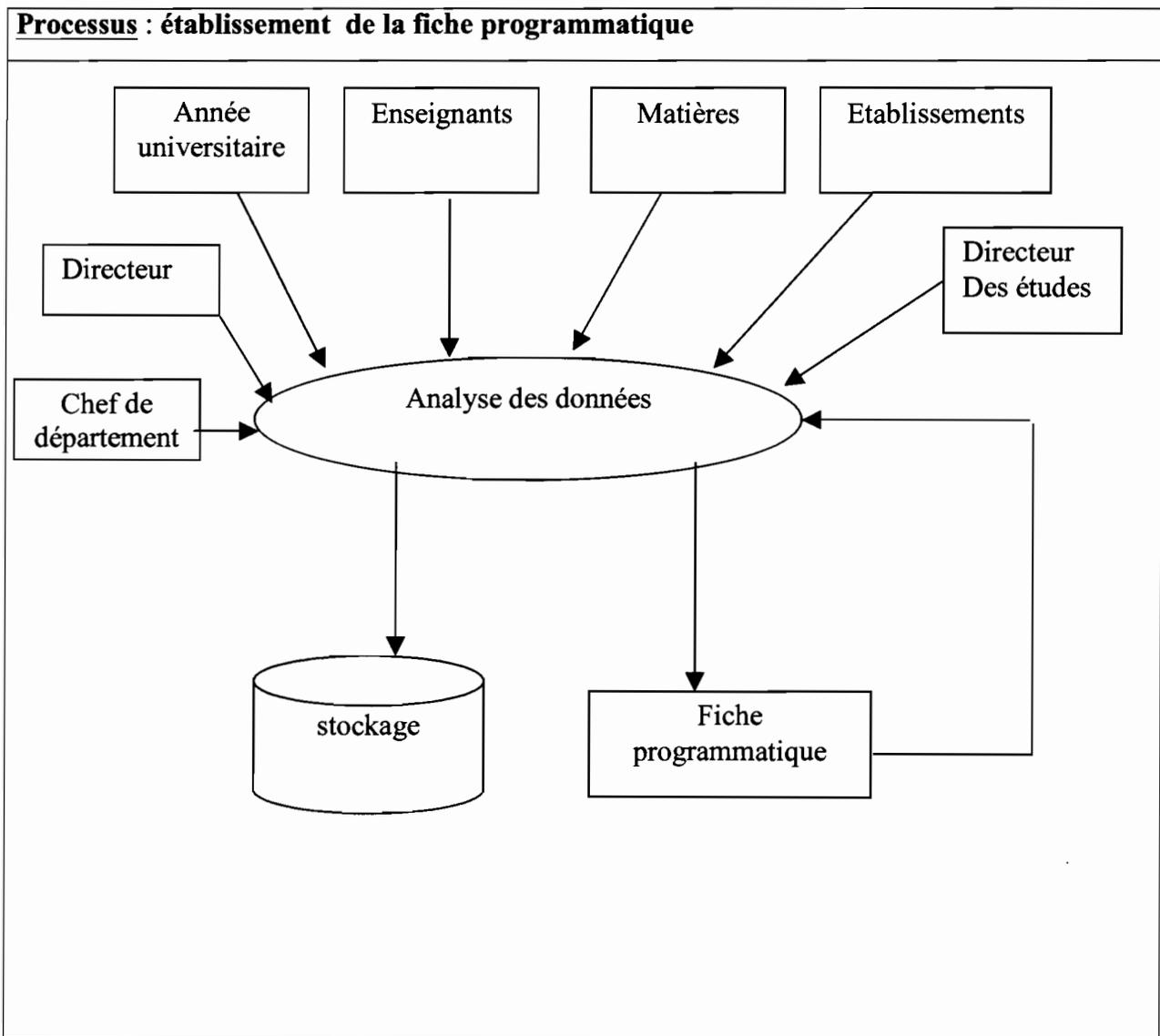


Figure 4 : scénario d'établissement de la fiche programmatique.

III.3.2. LE PROCESSUS D'ETABLISSEMENT DE LA FICHE FINANCIERE

Ce scénario est réalisé le plus souvent après une mission d'enseignement effectuée par des enseignants missionnaires ou vacataires. Il concerne aussi les enseignants permanents qui font des heures supplémentaires.

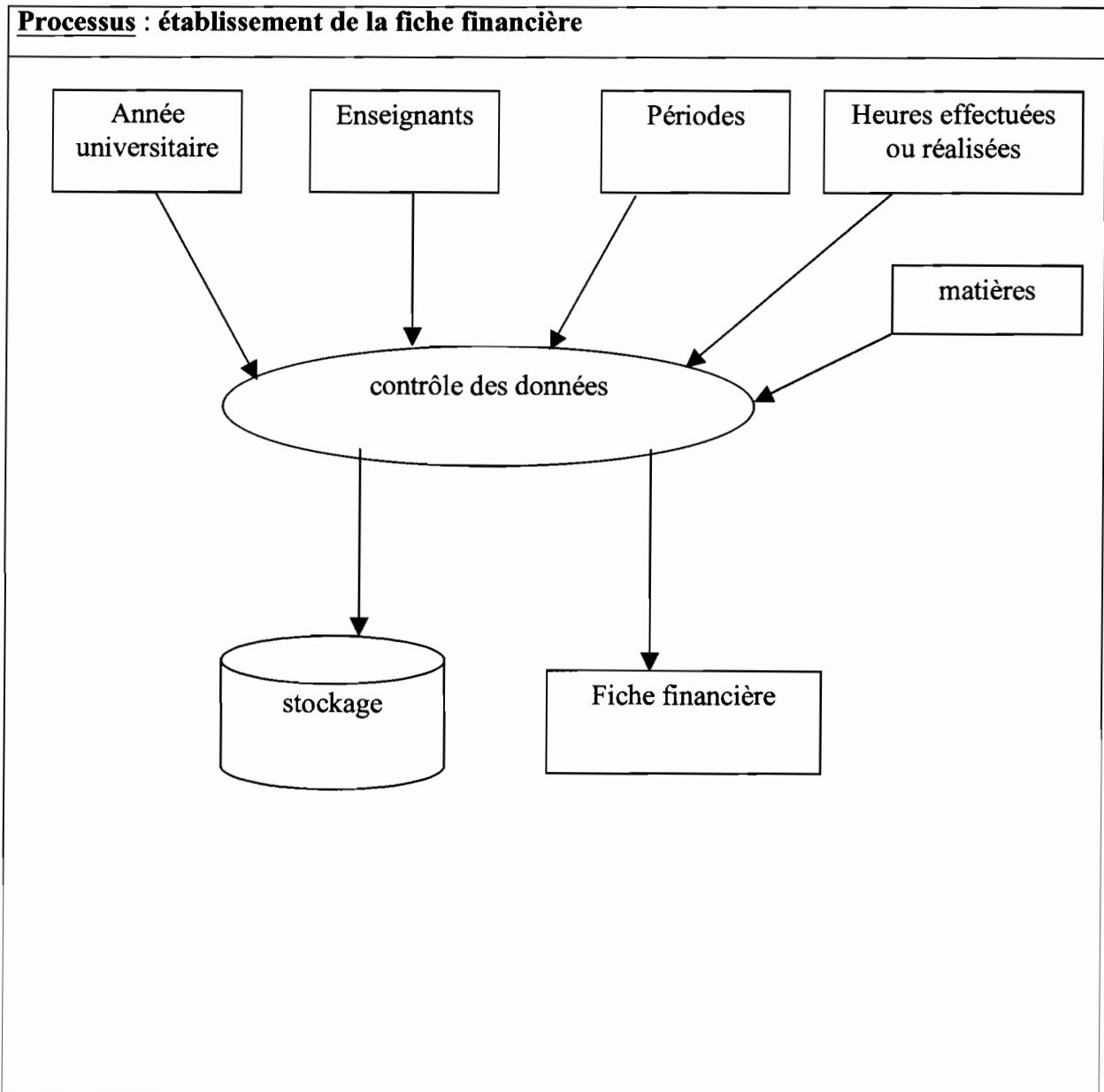


Figure 5 : scénario d'établissement de la fiche financière

III.3.3. LE PROCESSUS DE PROPOSITION DE L'EMPLOI DU TEMPS HEBDOMADAIRE

A partir des contraintes des enseignants, des salles, des fiches programmatiques et d'autres acteurs figurant sur la figure 6, on propose un emploi du temps hebdomadaire.

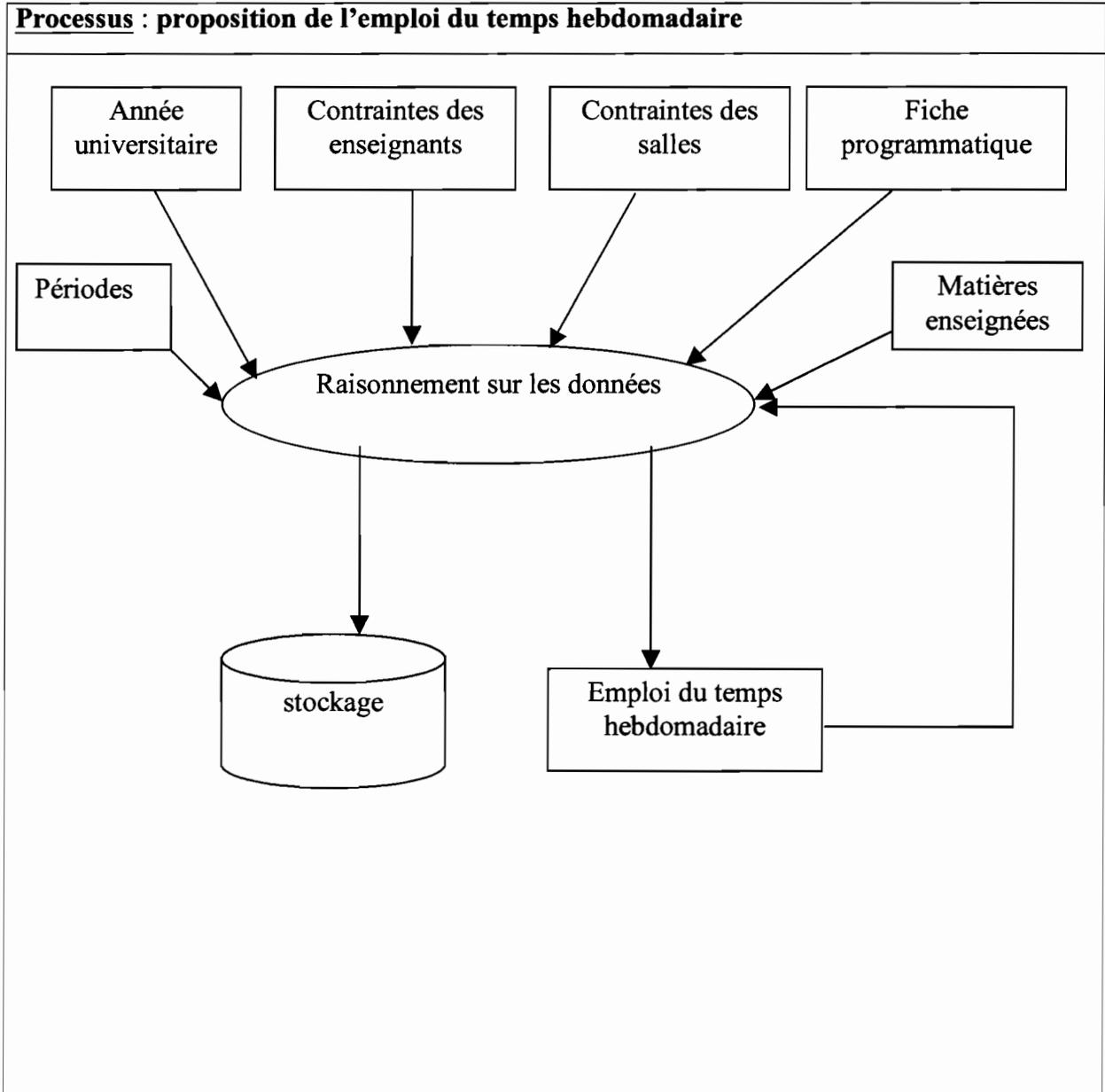


Figure 6 : scénario de proposition d'emploi du temps hebdomadaire

III.4. LE DEPLOIEMENT DE L'APPLICATION SUR LE RESEAU

Les réseaux d'ordinateurs permettent de répartir les données et les traitements sur des sites distincts. Cette conception de répartition permet de mieux adapter le système informatique (logiciel et matériel) à la réalité du fonctionnement du système d'information. Elle offre de nombreux avantages sur la disponibilité des données, la résistance aux pannes et la sécurité des données.

Nous limiterons notre choix au niveau des bases de données distantes et des bases de données réparties.

III.4.1. L'APPROCHE BASE DE DONNEES DISTANTE

Cette approche permet d'accéder à la base de données par une connexion réseau (figure 7). Elle suppose la mise en œuvre d'un réseau avec un serveur de données et de procédures permettant d'atteindre la base de données.

Une base de données distante est une base de données située sur un ordinateur distant (serveur) différent de celui qui traite les programmes d'application de l'utilisateur. En général, l'utilisateur doit connaître la localisation de la base de données distante qui doit être précisément identifiée pour accéder aux données distantes. Un service de communication de données est nécessaire afin d'accéder à la base.

A cet effet, les machines (les postes de travail) des utilisateurs qui seront sur le réseau local et qui pourront accéder à Internet pourront avoir l'interface de l'application en connaissant le nom du serveur et l'emplacement de l'application. En ce qui concerne l'accès à l'interface de l'application par Internet, cela suppose de disposer une connexion du réseau local à Internet. Ce modèle suppose un fonctionnement associant plusieurs ordinateurs reliés entre eux par un réseau. Ce fonctionnement repose sur un dialogue (émission de requêtes, appels de service par le client et émission de résultats par le serveur) régi par un protocole qui assure le bon déroulement des échanges. Il permet de répartir les services fournis par une application et non l'application elle-même.

Ainsi, avec l'interface, l'utilisateur pourra faire ses requêtes qui seront transmises au système de gestion de bases de données qui sert de pont entre l'application et la base de données.

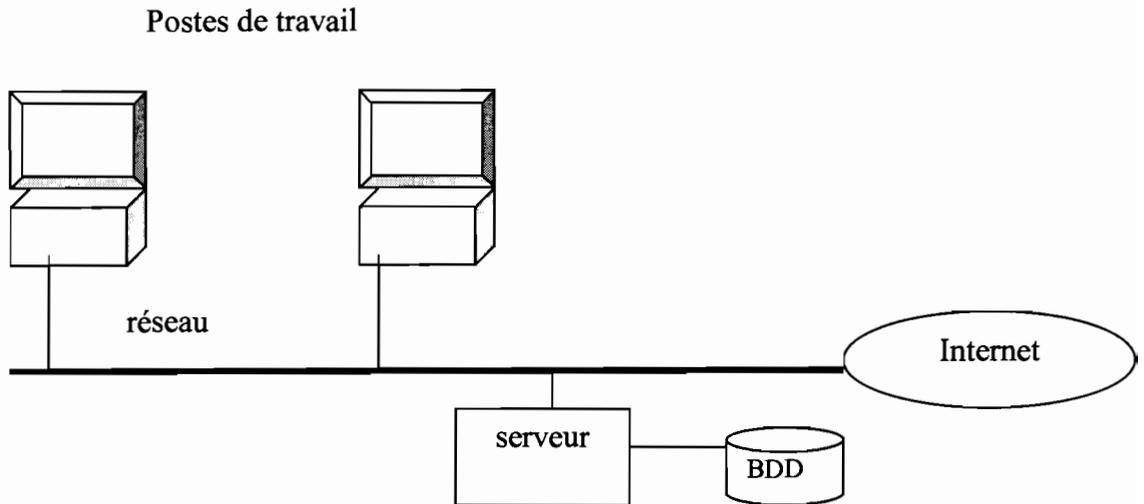


Figure 7 : schéma d'une approche de base de données distante

III.4.2. L'APPROCHE BASE DE DONNEES REPARTIE

C'est aussi une manière de répartir une application pour les utilisateurs. Une base de données répartie est un ensemble de bases de données coopérantes qui résident sur différentes machines, que nous appelons sites, interconnectés par un réseau de communication. Pour cet approche, l'utilisateur travaille comme s'il avait l'ensemble des bases de données coopérantes sur son ordinateur. Cela signifie que pour l'utilisateur, l'accès aux données pourrait être réparti sans qu'il ait à connaître ni à gérer cette répartition.

L'approche base de données répartie peut être définie aussi comme la communication de plusieurs réseaux ayant chacun sa base de données.

La gestion d'une base répartie nécessite sur chaque site l'installation des composants systèmes matérialisés par la figure 7.

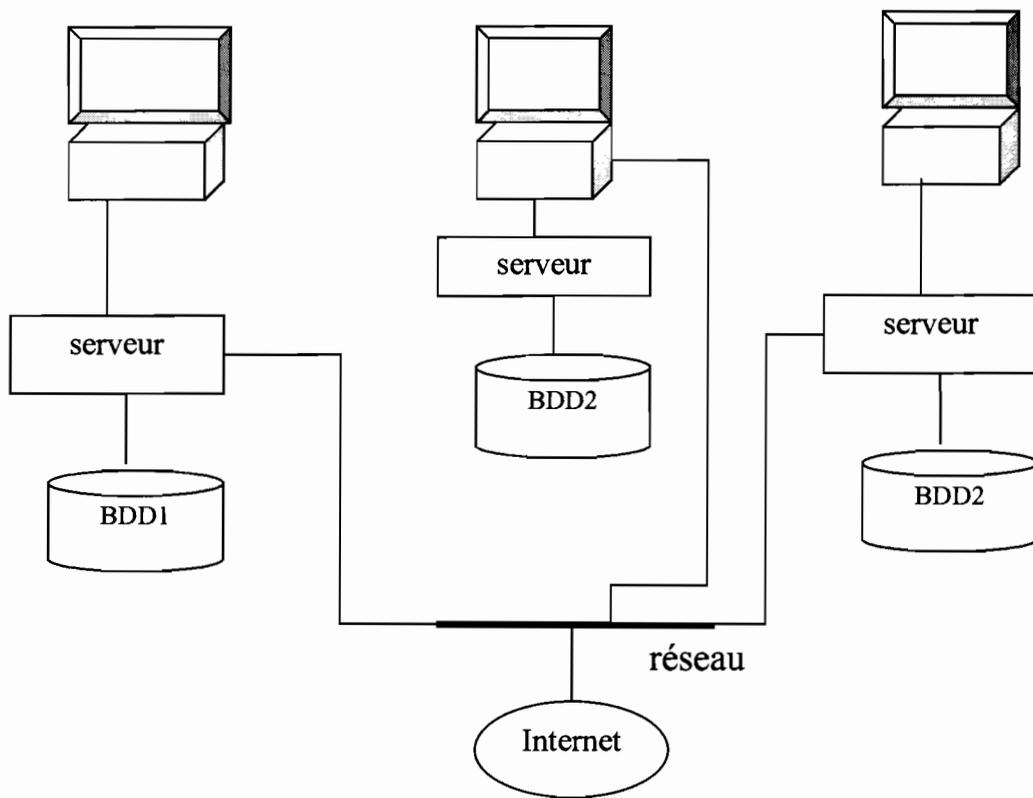


Figure 8 : schéma d'une approche de base de données répartie

L'approche répartie sera retenue par rapport à l'approche distante du fait :

- de la localisation des données pour chaque directeur des études;
- de la confidentialité des informations de chaque établissement ;
- de la simplification de l'installation et de la réduction du coût des communications car les échanges entre sous-systèmes étant relativement peu fréquents, peuvent être réalisés par des organes lents, l'information ainsi échangée étant en général plus synthétique;
- de l'avantage découlant de la structure modulaire : plus grande facilité de mise au point et d'intégration, commodité de modification et d'extension ;
- plus grande disponibilité dans la mesure où un organe défaillant peut être remplacé par un organe assurant une fonction équivalente, ou possibilité de fonctionnement dégradé en cas de défaillance d'un sous-système ;
- possibilité de décentralisation des données et des décisions, et accroissement de l'autonomie des sous-systèmes.

IV. L'ARCHITECTURE DU SYSTEME

L'architecture d'un système est la description de la structure globale d'une application ou d'un système informatique.

Cette partie sera composée de la description de l'architecture globale du système, de l'architecture de l'application et du moteur d'inférence.

IV.1. ARCHITECTURE GLOBALE DU SYSTEME

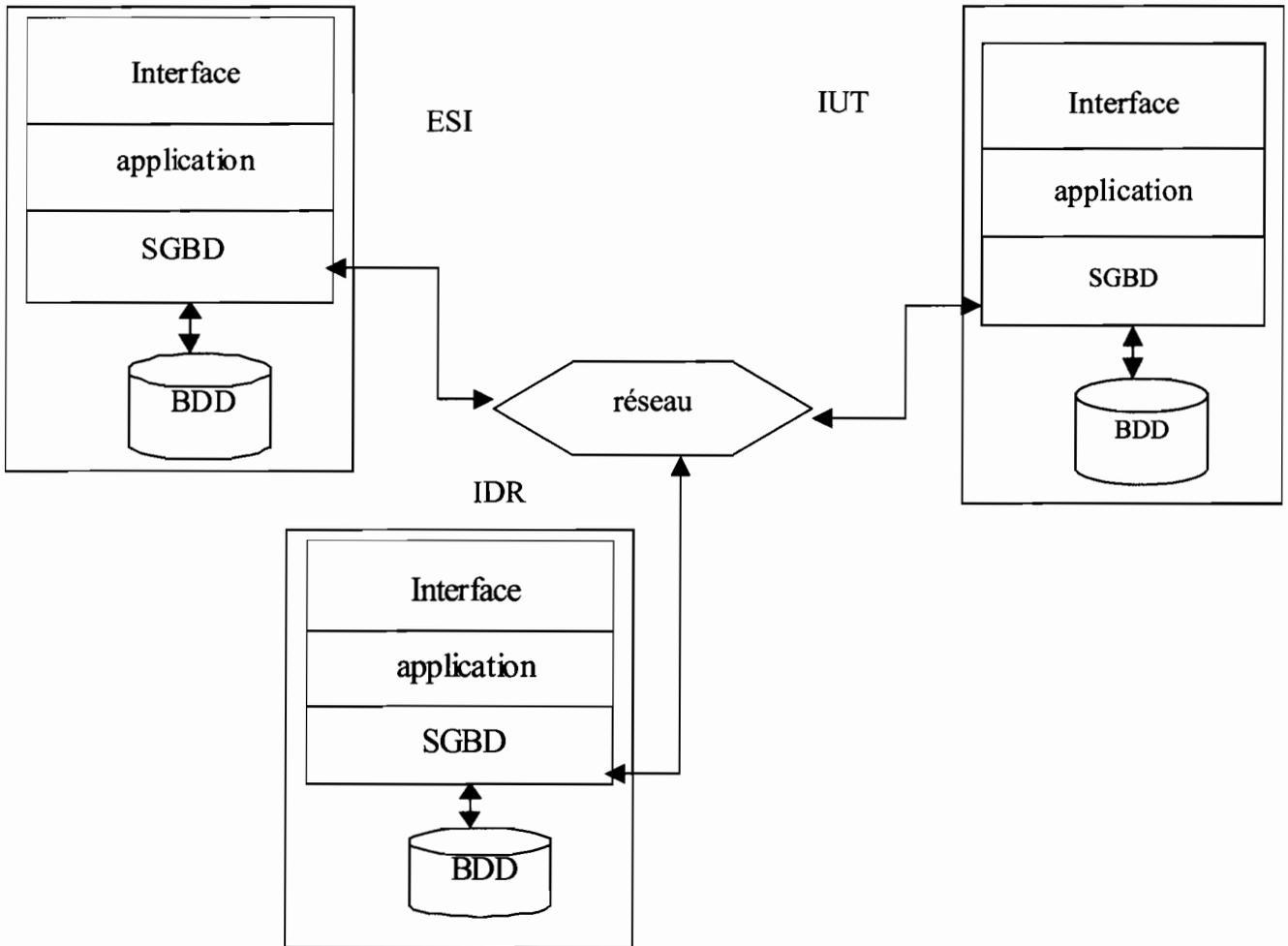


Figure 9 : schéma de l'architecture générale de notre système

Deux propriétés importantes d'une base de données répartie sont l'homogénéité et l'hétérogénéité.

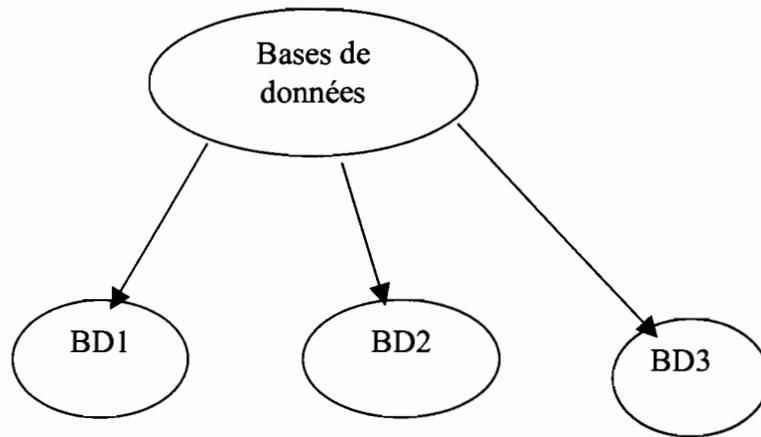


Figure 10 : *Base homogène (division d'une base de données en bases de données locales)*

Une base de données répartie homogène résulte en général d'une approche de conception descendante où l'on la conçoit à partir d'une base de donnée centralisée. Elle facilite l'ajout d'un nouveau site dans le réseau.

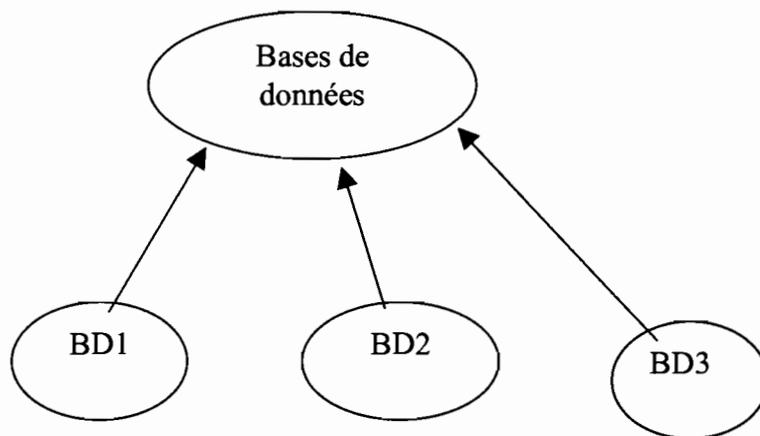


Figure 11 : *Base hétérogène (intégration des bases de données existantes en une base de données)*

La caractéristique d'une base de données répartie hétérogène est d'intégrer des bases de données locales gérées par des systèmes de gestion de base de données qui peuvent être différents.

Cette approche qui permet l'intégration des bases de données existantes et indépendantes sans avoir à créer une base de données répartie complètement nouvelle, sera l'approche à maintenir.

Une base de données répartie hétérogène vise l'objectif d'indépendance des systèmes de gestion de bases de données locaux. C'est la seule approche qui permette d'intégrer en une

seule base globale des bases de données préexistantes gérées par des systèmes de gestion de base de données différents tels que relationnel et réseaux.

L'avènement du modèle relationnel et la standardisation du langage des requêtes SQL ont favorisé le développement des bases de données hétérogènes. Cette tendance est bien illustrée par des systèmes de gestion de base de données relationnels populaires tels que ORACLE et INGRES qui proposent des solutions intéressantes à la coopération de bases de données relationnelles gérées par différents SGBD.

ORACLE /STAR et INGRES/STAR dérivés respectivement d'ORACLE et de INGRES sont des SGBD répartis hétérogènes . Ils permettent d'intégrer des données locales gérées uniquement par des SGBDR.

Voici un exemple décrit dans le modèle relationnel d'INGRES et manipulé au moyen du langage SQL implanté par INGRES. Chaque traducteur ou passerelle convertit les requêtes et données SQL en données et requêtes du SGBD local.

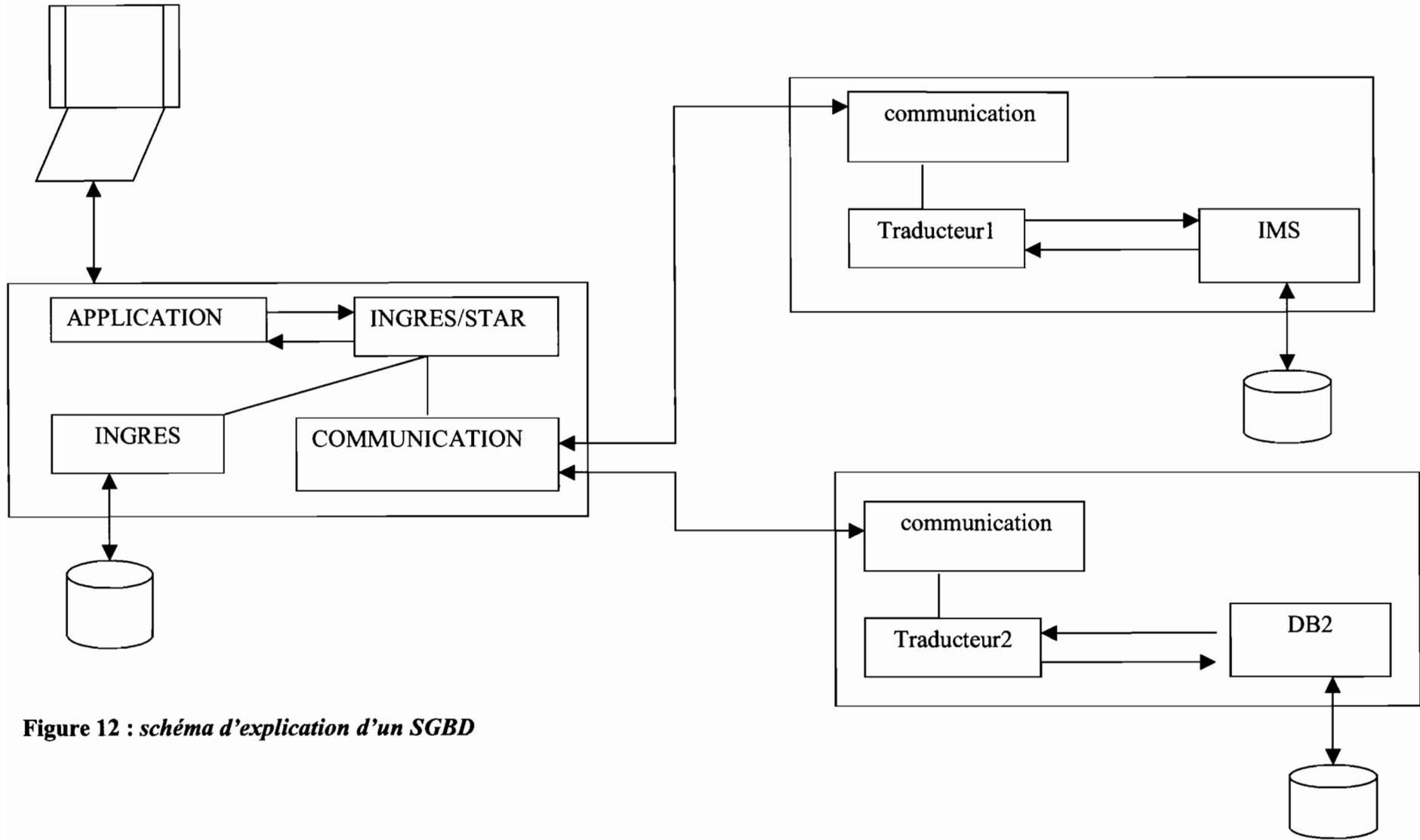


Figure 12 : schéma d'explication d'un SGBD

IV.2. ARCHITECTURE DU SYSTEME DE GESTION DES ENSEIGNEMENTS

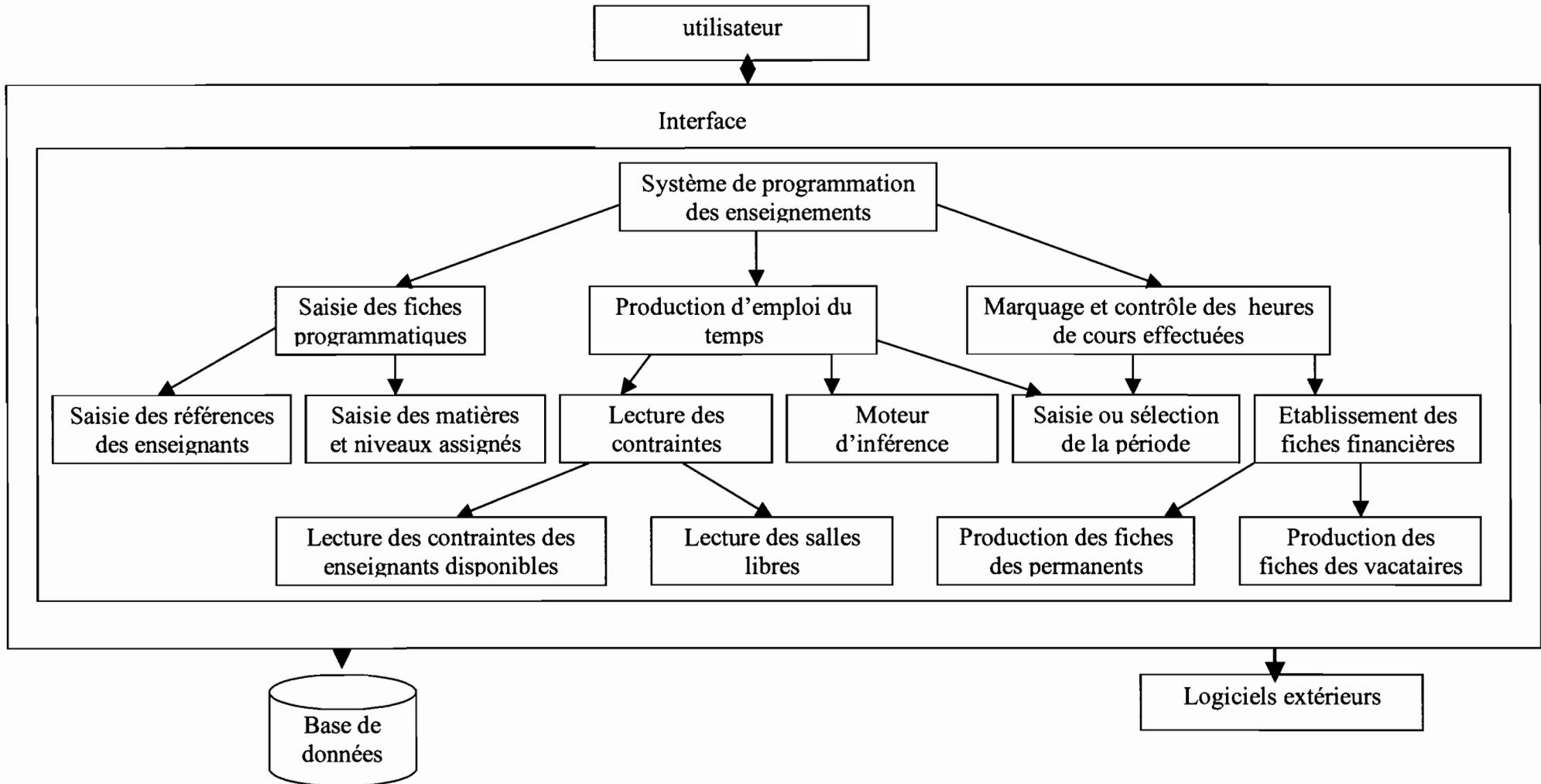


Figure 13 : architecture du système de gestion des enseignements

IV.3. LE MOTEUR D'INFERENCE

Le moteur d'inférence est la partie active du système. C'est un programme général qui va mettre en œuvre la connaissance contenue dans la base. Ce moteur de développement recouvre les procédures isolées ou non dans un module. Il fonctionne selon une stratégie spécifique comprenant la restriction, le filtrage et la résolution des conflits et peut être matérialisé par le schéma de la figure 14 ci- après.

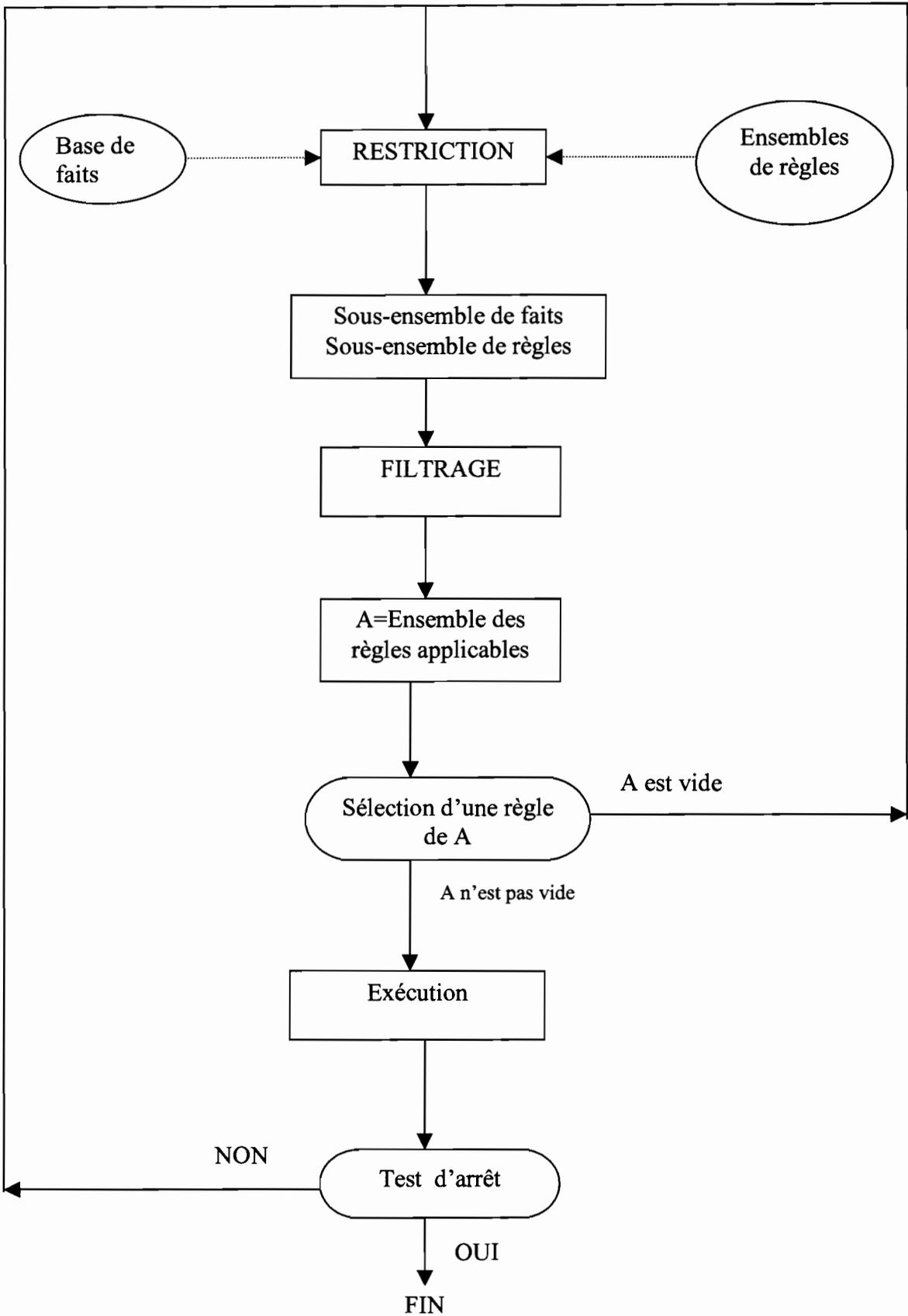


Figure 14 : le fonctionnement du moteur d'inférence

IV.3.1. LA RESTRICTION OU SELECTION

Le module de sélection doit prévoir ce qu'il faut faire si l'ensemble des règles applicables est vide. Si la restriction est trop forte on peut envisager d'essayer à nouveau avec une restriction moins forte. Si toutes les règles ont été essayées, on est obligé de décider de la fin des opérations même si le test d'arrêt n'est pas satisfait.

Dans le cas où A contient plusieurs règles, le moteur peut être conçu pour les exécuter les unes après les autres. Dans l'autre cas (voir figure 14), après exécution d'une règle, on repart au début sans épuiser l'ensemble A.

L'opération de filtrage est souvent longue en temps de calcul. Pour alléger le fardeau que constitue le nombre de tests à effectuer, on peut décider que seule une partie des règles ou des faits va être filtrée. Une phase de restriction intervient en ce moment avant la phase de filtrage. Elle a pour rôle de restreindre à partir des faits disponibles les règles qui sont susceptibles d'aboutir au résultat.

Elle se fait sur les bases des faits et des données et sélectionne à priori les règles qui seront examinées. Les principales règles pour la réalisation d'emploi du temps hebdomadaire sont les suivantes :

- La période concernée est une période incluse dans l'année universitaire ;
- Les jours à programmer sont des jours ouvrables ;
- Ne sont concernés pour une période donnée que les enseignants qui sont disponibles ;
- Un enseignement ne peut se dérouler que s'il existe des salles libres .

IV.3.2. LE FILTRAGE

Le filtrage est une étape du moteur d'inférence qui consiste à déterminer l'ensemble des règles dont les conditions satisfont l'état courant de la base de données. C'est une phase qui intervient après la restriction pour diminuer encore les règles à explorer pour atteindre le but. Le filtrage permet d'avoir les règles et les faits qui répondent à ces règles qui sont tout susceptibles d'aboutir à une solution.

Il consistera dans la réalisation de l'emploi du temps à regrouper :

- les enseignants disponibles
 - par jour ;
 - par classe ;
 - par horaire ;
- les salles libres
 - par jour ;
 - par horaire ;
 - par type.

IV.3.3. LA RESOLUTION DES CONFLITS

La résolution de conflits est une étape qui survient après le filtrage au niveau de l'exécution. Elle permet de choisir la règle à exécuter parmi une multitude de règles qui peuvent aboutir au résultat. En d'autres termes, Ce sont des critères dont il faut tenir compte pour départager des règles en conflits.

Elle consistera à sélectionner parmi les enseignants qui sont disponibles à la même date, aux mêmes horaires et enseignant la même classe celui qui sera programmé. Elle se fera selon le statut de l'enseignant, selon le pourcentage des heures effectuées des matières en conflits, selon la précedence des matières et aussi dynamiquement en cas d'impasse. L'utilisateur pourra faire des modifications et dans ce cas, ce sera au système de les vérifier et de les valider.

IV.3.4. STRUCTURE DU MOTEUR D'INFERENCE

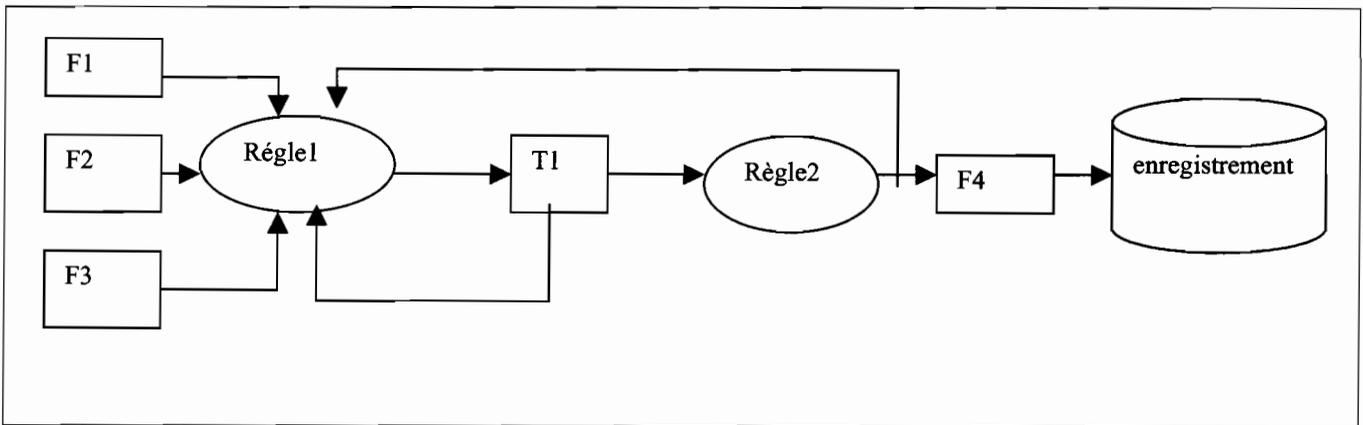


Figure 15 : structure du moteur d'inférence

NOTATION	DESCRIPTION
F1	Saisie ou sélection de la période
F2	Enseignants disponibles
F3	Salles disponibles
Règle1	Si F1, F2, F3 réalisés alors T1
T1	Regroupement des enseignants disponibles par classe, par jour, par horaire
Règle2	Résolution de conflits
F4	Emploi du temps hebdomadaire

Tableau 2 : explication de la structure du moteur d'inférence

IV.3.5. ALGORITHME DE RAISONNEMENT POUR RESOUDRE LES CONFLITS

A) Programmation des enseignements dans les classes en fonction des disponibilités des enseignants et des salles ;

B) choisir une classe et analyser les matières à enseigner par les enseignants en fonction de A) ;

si (volume_horaire_matière est fini) ou (si les matières précédent une matière ne sont pas démarrées) alors éliminer cette matière de la sélection

sinon

regrouper ses matières en fonction des horaires de disponibilité des enseignants qui enseignent ses matières

si des enseignants ont les mêmes horaires de disponibilité alors regarder leur statut

s'ils ont le même statut alors

si les pourcentages des heures de cours effectués sont les mêmes choisir dynamiquement

sinon programmer le moins avancé

.fin si ;

Sinon

Programmer les missionnaires d'abord,

Ensuite les vacataires,

Et enfin les permanents.

Fin si

Fin si

Fin si

faire un rapprochement des horaires de disponibilité des enseignants et des salles

Procéder aux différentes affectations ;

Mise à jour des horaires de disponibilité des enseignants, salles et classes.

V. CHOIX DES OUTILS

La mise en œuvre d'un système informatique efficace ne dépend pas seulement de l'étude conceptuelle mais aussi du matériel de réalisation. Cette partie est très capitale pour le concepteur qui veut réaliser le système et pour l'utilisateur qui veut voir son système fonctionner avec les dernières technologies.

Nous choisirons dans cette partie la base de données, le langage d'implémentation, le réseau et le type de serveur à mettre en place.

V.1.CHOIX DE LA BASE DE DONNEES

Une base de données peut être définie comme une organisation cohérente de données permanentes et accessibles par des utilisateurs concurrents. Une base de données est dite cohérente si elle satisfait l'ensemble des contraintes sémantiques qui lui sont associées, ceci en présence de pannes matérielles ou logicielles. La gestion de la cohérence nécessite donc les mécanismes de contrôle d'intégrité et de reprise sur panne. La permanence ou persistance des données assure la gestion d'accès au disque (placement, index, etc...) et la gestion d'un cache pour augmenter les performances d'accès aux données. La concurrence d'accès en lecture ou écriture de la base doit être contrôlée par un mécanisme de gestion de transaction. Son rôle est d'assurer un degré multi-utilisateur élevé tout en maintenant la cohérence des données.

Il existe des bases de données objets et des bases de données relationnelles parmi lesquelles on retiendra la base de données à utiliser.

V.1.1. LA BASE DE DONNEES OBJETS

La première motivation pour les bases de données objet est le besoin d'accès à des objets persistants par des programmes écrits dans un langage à objets. Une seconde motivation, peut-être plus importante, vient des nouveaux besoins exprimés par des domaines d'applications plus complexes comme l'ingénierie ou le multimédia.

Les bases de données objets visent à atteindre trois objectifs complémentaires :

- réduire, voire éliminer le dysfonctionnement entre langages de programmation et langages de bases de données ;
- supporter directement des objets arbitrairement complexes grâce à un modèle à objets ;
- partager le code réutilisable des applications.

Les bases de données objets doivent permettre l'accès (en création, lecture, écriture, destruction) transparent à des objets persistant à partir d'un langage objet. Il n'y a pas de modèle unique de bases de données objets. Les modèles de bases de données objets sont issus des langages à objets et des modèles de données sémantiques. L'objectif commun à ces

modèles est de pouvoir capturer toute la sémantique des données en termes de concepts tels que entité, association, agrégation et généralisation.

Plus spécifiquement, la base de données objet apporte des avantages. Elle permet de prendre en compte la dynamique associée aux objets sous forme d'opérations stockées avec des objets et par là une réponse partielle au problème de réutilisabilité du code des programmes. De plus, la possibilité d'incrémenter la connaissance stockée dans la base par ajout de nouveaux types d'objets et d'opérations associées rend le SGBD extensible.

V.1.2. LA BASE DE DONNEES RELATIONNELLE

La base de données relationnelle permet d'organiser les données en tables. Cette théorie a pour but d'éliminer les comportements anormaux des relations lors des mises à jour. Elle permet aussi d'éliminer les données redondantes et de mieux comprendre les relations sémantiques entre données.

Le modèle relationnel inventé par E.F Codd à IBM-San José en 1970 peut être décrit par les trois propriétés suivantes:

- les structures de données sont simples et se construisent à partir de la théorie des ensembles. Ce sont des tables à deux dimensions dont chaque élément appartient à un ensemble de valeurs appelés relations. Une colonne d'une relation appelée attribut, est définie sur un même domaine. Une ligne d'une relation, appelée n-uplet ou tuple, relie logiquement les éléments d'informations fournis par les valeurs d'attributs. Toute l'information de la base de données doit être représentée explicitement par des valeurs dans les tables. En particulier, il n'y a pas de pointeurs interrelations visibles par l'utilisateur ;
- un ensemble d'opérateurs appliqués à des relations, constituant l'algèbre relationnelle, permet la définition, la recherche et la mise à jour des données. Chaque opérateur prend une ou deux relations en argument et produit une relation. L'algèbre relationnelle comprend les opérateurs classiques de la théorie des ensembles (Produit cartésien, Union, Intersection, différence) et des opérateurs permettant de décomposer des sous-ensembles d'une ou de plusieurs relations (projection, restriction, jointure et division) ;
- un ensemble de contraintes d'intégrité sémantiques définit les états cohérents de la base de données.

L'avantage majeur du modèle relationnel est sa facilité à assurer l'indépendance complète entre la description logique de données (en terme relationnel) et physique (en terme de fichier et liaison entre fichiers). Cette indépendance physique a permis le développement de langages de définition et de manipulation de données de haut niveau, appelés langages de requêtes, au dessus de l'algèbre relationnelle. Ces langages sont typiquement assertionnels, c'est-à-dire basés sur la logique des prédicats, et libèrent le programmeur de la spécification des chemins d'accès aux données. En conséquence, l'optimisation des requêtes de manipulation de données peut être entièrement automatisée.

Nous retiendrons le modèle relationnel qui connaît un succès certain en raison de:

- sa simplicité et aussi des langages de manipulation de données associés qui sont nombreux et puissants;
- la facilité d'administrer une base de données relationnelle (sécurisation, répartition...);
- la taille du projet ;
- du langage envisagé pour l'implémentation.

V.1.3. TRADUCTION D'UN MODELE ENTITE – ASSOCIATION EN MODELE RELATIONNEL

Compte tenu de l'importance que l'on a accordée à la présentation de l'approche entité - association dans la modélisation des données d'un système d'information, on présentera de manière plus schématique et condensée le modèle relationnel d'autant plus que ces deux approches sont équivalentes.

L'intérêt qu'il y a de présenter le modèle relationnel découle entre autres de deux considérations :

- Le modèle relationnel présente la structure des données dans une forme tabulaire qui est proche de la manière dont l'information est en définitive stockée sous formes de fichiers, ce qui permet d'évaluer la place prévisible pour la stocker.
- Vu le développement des « systèmes de gestion des bases de données relationnels », et leur intérêt du point de vue de l'utilisateur final qui dispose ainsi des langages tels SQL permettant d'interagir avec l'information gérée, la connaissance de ce modèle est nécessaire pour le gestionnaire.

V.1.3.1. LES CONCEPTS DE LA MODELISATION RELATIONNELLE

- **Relation** : dans le modèle relationnel, l'information est vue sous la forme de « tables » comportant des colonnes qui ne sont rien d'autres que les attributs du modèle entité – association, et des lignes correspondant aux occurrences d'une entité ou d'une association. Une table a un nom, celui de la relation. Un système est représenté par un ensemble de relations.
- **Clé d'une relation** : une relation comporte une clé formée d'un ou plusieurs attributs ; il s'agit de l'identifiant dont il a été question dans la modélisation des données. Les attributs identifiants sont soulignés dans la relation.

V.1.3.2. LES REGLES DE LA TRADUCTION

Comme l'indiquent les trois règles essentielles énoncées ci-après, la traduction d'un modèle entité – association en un modèle relationnel est simple.

- Règle 1 : toute entité est traduite en une relation avec les mêmes attributs et comme clé l'identifiant de l'entité. Le schéma de la figure 16 qui suit résume cette règle :

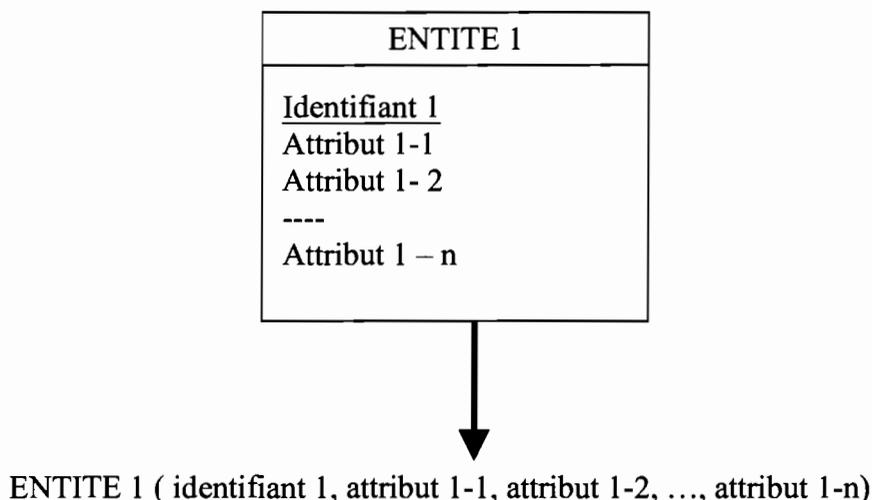


Figure 16 : schéma d'explication de la règle 1 de passage d'une entité-association à un modèle relationnel

- Règle 2 : Toute association ayant une cardinalité 0,1 ou 1,1 avec une entité se traduit par l'ajout dans la relation équivalente à cette entité, avec le statut d'attribut, des identifiants des autres entités associées. Cette règle est exprimée schématiquement par la figure qui suit :

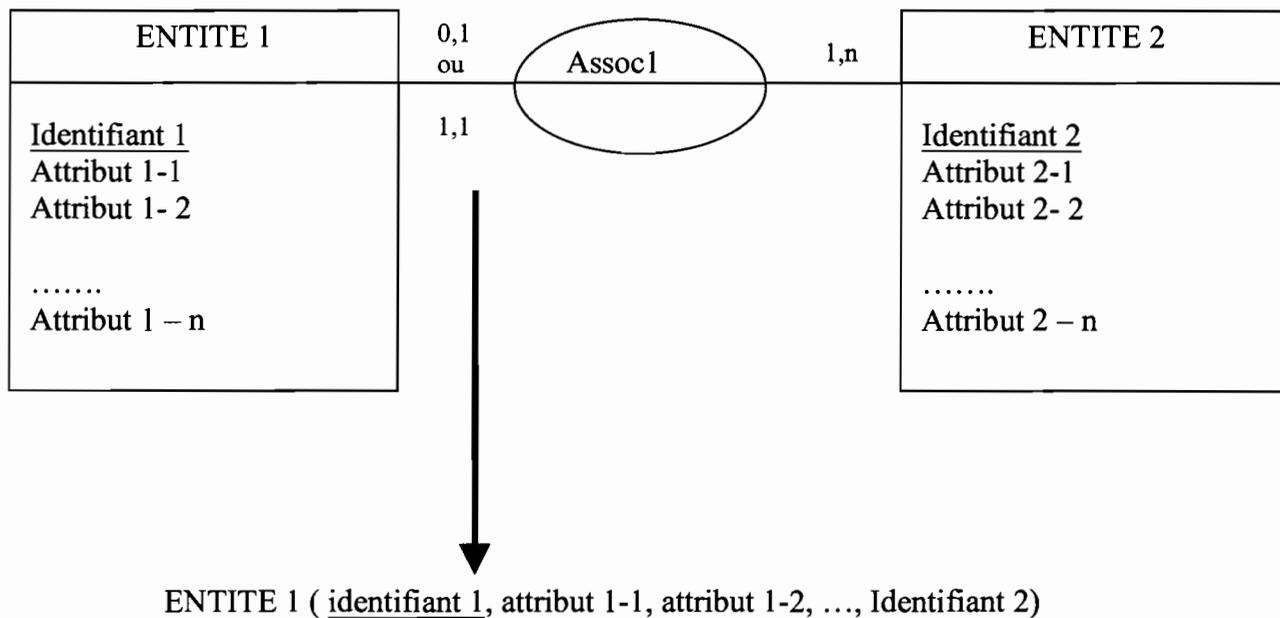


Figure 17 : schéma d'explication de la règle 2 de passage d'une entité-association à un modèle relationnel

- Règle 3 : une association sans une cardinalité 0,1 ou 1,1 est traduite par une relation dont la clé est l'ensemble des identifiants des entités associées, à laquelle on ajoute les éventuels attributs de l'association.

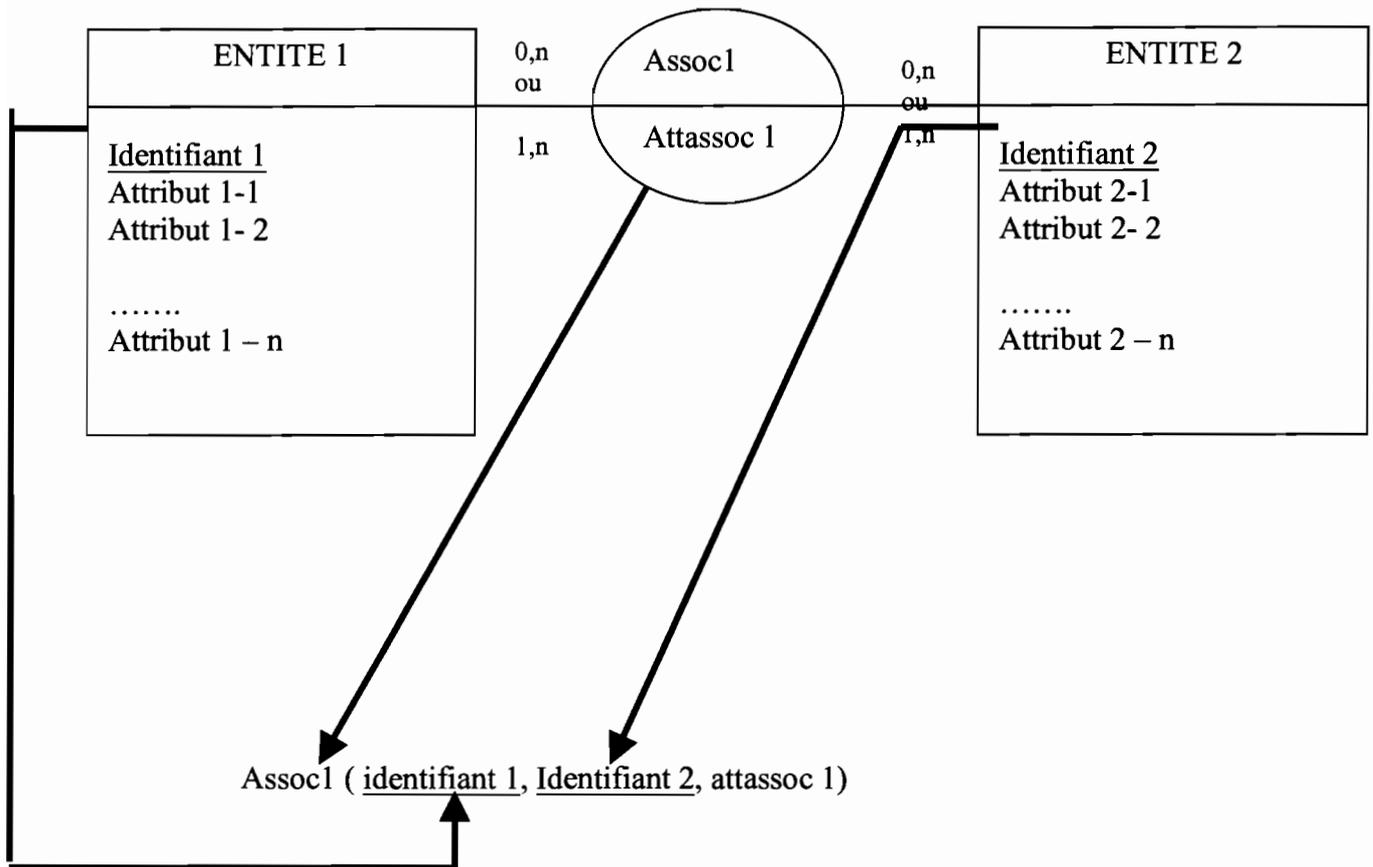


Figure 18 : schéma d'explication de la règle 3 de passage d'une entité-association à un modèle relationnel

V.1.3.3. APPLICATION DE CES REGLES A NOTRE SYSTEME

Si l'on se rapporte à notre modèle entité – association représenté par le modèle conceptuel de données (figure 3), on aboutit au modèle relationnel équivalent en leur appliquant ces trois règles.

LES TABLES DE LA BASE DE DONNEES

STATUT

(CODE STATUT , LIBELLE_STATUT);

GRADE

(CODE GRADE, LIBELLE_GRADE, VOLUME_HOR_STATUT_AN,
COUT_HORAIRE);

FONCTION
(CODE FONCTION , LIBELLE_FONCTION) ;

ANNEE_ACADEMIQUE
(ANNEE ,);

POLE_D_ENSEIGNEMENT
(CODE POLE, LIBELLE_POLE);

SEMAINE
(CODE SEM ,DEB_SEM, FIN_SEM);

MATIERE
(CODE MAT ,CODE_POLE ,LIBELLE_MAT);

JOURS
(JOUR , CODE_SEM);

HORAIRES
(PLAGE_HORAIRE ,JOUR);

ENSEIGNANT
(CODE_ENS , CODE_GRADE, CODE_STATUT , CODE_FONCTION , NOM_ENS ,
PRENOM_ENS , ADRESSE_ENS , ADRESSE_ELECT ,DIPLOME_ENS ,
ETABLISSEMENT_ENS);

CLASSES
(NIVEAU , CYCLE , ETABLISSEMENT) ;

SALLES
(CODE_SALLE , PLAGE_HORAIRE , LIBELLE_SALLE , BATIMENT , TYPE_SALLE ,
CAPACITE_SALLE);

ENSEIGNE_CLASSE
(CODE_ENS , NIVEAU , CYCLE , ETABLISSEMENT);

MATIERE_EST_AFFECTE
(CODE_ENS ,CODE_MAT , HEURE_COURS , HEURE_TP , HEURE_TD);

ENSEIGNE_ANNEE
(CODE_ENS , ANNEE);

MAT_CLASSE
(CODE_MAT , NIVEAU , CYCLE , ETABLISSEMENT , VOLUME_COURS ,
VOLUME_TD , VOLUME_TP);

RESERVE
(CODE_ENS ,PLAGE_HORAIRE);

EST_PROG2
(CODE MAT , PLAGÉ HORAIRE);

LIBRE
(PLAGÉ HORAIRE , CODE SALLE);

APPARTIENT
(CODE SEM , CODE_POLE , ANNEE);

V.2.CHOIX DU LANGAGE

V.2.1. LANGAGE PROCEDURAL (CLASSIQUE) OU LANGAGE DECLARATIF (D'INTELLIGENCE ARTIFICIELLE) ?

La question se pose de réaliser cette application avec des outils de l'informatique classique. Faisant abstraction du langage, nous dirons qu'il faut opposer la programmation procédurale qui utilise les langages comme FORTRAN, PASCAL, C et les autres à la programmation dite déclarative de type PROLOG.

La programmation procédurale repose sur la description complète d'un algorithme et qu'elle impose le cheminement entre les différentes fonctions et procédures du programme, c'est-à-dire que le programmeur prévoit (en principe) l'enchaînement des fonctions et procédures.

De plus les langages de quatrième génération (L4G) font partie des outils classiques que nous rencontrons aujourd'hui presque partout. Ces langages se sont développés à partir des architectures d'infocentre pour traiter et analyser les données extraites des bases. Un langage de quatrième génération est un langage spécifique spécialisé dans le traitement des données intégrant à la fois un accès type SQL aux tables, des facilités de contrôle, d'affichage, de saisie interactive et d'édition de rapports. Un L4G est donc indépendant de tout langage natif, type C, Cobol ou Pascal. Il dispose d'une interface graphique pour composer et éditer des fenêtres d'écrans et pour dialoguer avec l'utilisateur. L'interface graphique est aussi utilisée pour générer des rapports à partir notamment des tables relationnelles.

Au cœur même du L4G, un langage de programmation événementielle permet de définir des procédures déclenchées suite à un événement. Les constructions du langage supportent la déclaration des variables, des calculs et assignation, les tests du type IF et CASE, les boucles LOOP et FOR EACH, le traitement des erreurs. Un L4G permet aussi l'appel des fonctions externes.

Au contraire dans la programmation déclarative comme dans PROLOG, les faits et les règles sont apportés en vrac, ensuite l'interpréteur ou le moteur d'inférence aménage lui-même son cheminement parmi les règles. Les règles et les procédures sont données indépendamment de leur enchaînement.

Le premier avantage de la programmation déclarative réside donc dans l'apport et le retrait des règles. Le deuxième repose sur le fait qu'un cheminement de procédure n'ayant pas

été conçu pour répondre à une question précise, le programme peut répondre à de nombreuses questions. Ceci laisse une plus grande liberté à l'utilisateur.

Certes, la programmation déclarative a des avantages. Mais il faut noter que le temps d'exécution (qui est grand du fait du parcours des règles) reste un handicap principal de la programmation déclarative. Si bien qu'il ne faut pas hésiter, quand le traitement d'un problème relève d'un algorithme bien défini et isolable, à utiliser un langage classique.

A travers ces données, nous nous baserons sur les langages déclaratives pour réaliser le moteur d'inférence de notre système et sur les langages classiques pour implémenter le reste de l'application. PROLOG sera utilisé pour le moteur d'inférence. Delphi et Visual Basic qui sont disponibles sur les machines du par machine de l'U.P.B nous limitera notre choix parmi les L4G et de décider entre ses deux langages.

V.2.2. VISUAL BASIC OU DELPHI ?

Delphi et Visual Basic sont les deux (2) langages de développement avec lesquels nous comptons développer notre application. Ces deux langages sont des langages de quatrième génération et offrent une facilité d'utilisation des outils (librairie, bibliothèque) et de création de logiciels dédiés à Internet ou à l'intra net.

Delphi est un langage orienté objet. La solution de Delphi est bien plus modulable que la solution Visual Basic.

- Du point de vue des fonctions qui doivent être partagées sous Visual Basic, ces fonctions doivent être placées dans la partie globale du fichier local de code ou dans un fichier global *.BAS. Ces fonctions sont en ce moment accessibles à l'entièreté du projet. Delphi permet à la fonction d'être placée dans un fichier local (ou dans un *.DCU, fichier d'unité de Pascal) et doit être explicitement déclarée par les fichiers qui l'utilisent.
- Du point de vue de l'utilisation et de la création des bibliothèques ou DLL(Dynamic Library Link) , il faut noter que :
 - Visual Basic utilise les OCX (les OCX sont accompagnés d'un fichier binaire) qui fonctionnent comme fichiers étendus de bibliothèque de type antémémoire pour le fichier d'ordre (les fichiers d'ordre étant les fichiers OCX qui ont la forme des commandes).
 - Delphi n'a pas besoin de ce type de fichier qui ralentit la compilation.

Au-delà des facilités des deux langages, le langage de la programmation de Delphi, le Pascal objet, permet aux développeurs de réutiliser et personnaliser les fonctions dans une classe par l'intermédiaire d'un descendant (notion d'héritage). Un objet en Visual Basic est un OCX statique créée avec Visual Basic à côté du projet et distribué par l'application. Sous Delphi, les objets font partie du code du programme tout comme de simples routines. Ils peuvent être créés en cours d'exécution du programme même , et modifiés simplement.

Avec Visual Basic, la création de descendant est impossible : si un OCX spécifique est insuffisant pour un utilisateur de Visual Basic, la seule alternative est d'établir ou d'acheter

un autre OCX. Si la notion d'objet existe en Visual Basic, la notion de descendance n'existe pas. De plus si l'usage de Delphi assure la portabilité de l'application sous l'environnement Windows comme celui de linux, ce n'est pas le cas de l'application au Visual Basic qui n'est portable que sur l'environnement Windows.

Visual Basic propose des Webclasses (composant Visual Basic résidant sur un serveur Web et répondant aux appels du navigateur) afin de créer des applications Web serveur. Ces Webclasses sont accompagnées d'un concepteur de page web au format DHTML permettant de programmer des pages Web interactives.

Delphi, avec Internet Express propose une suite de composants pour construire des clients dynamiques pouvant prendre en charge de nombreux utilisateurs et les volumes de données importants.

En résumé, nous présentons le tableau suivant :

Tableau comparatif de VB et de DELPHI		
Critères	VB	DELPHI
Langage	Evénementiel	Orienté objet
Internet	DHTML, Application IIS	Internet express
Portabilité	Windows	Linux ,Windows
Reutilisabilité	Limites aux ocx existant	Assuré par le concept d'héritage

Tableau 3 : Tableau comparatif de VB et de DELPHI

De la comparaison des deux langages (tableau 3), il apparaît plus avantageux d'utiliser DELPHI pour notre application.

V.3.CHOIX DU RESEAU (ETHERNET OU TOKEN RING)

V.3.1. LES COMPOSANTES DU RESEAU LOCAL

Le terme de système informatique réparti ou distribué est utilisé de manière courante pour désigner des ensembles informatiques constitués d'unités de traitement et de stockage interconnectées par un système de communication. Sur ces unités s'exécutent les programmes qui coopèrent à la réalisation d'une tâche commune ou partagent des ressources communes. Nous pouvons rappeler les trois fonctions assurées par le réseau :

- permettre aux correspondants qui sont reliés de dialoguer entre eux ;
- permettre le déroulement simultané de plusieurs dialogues en partageant entre eux l'ensemble des ressources du réseau (lignes, commutateurs) ;
- éviter les interférences entre plusieurs dialogues distincts.

On désigne sous le terme de réseaux locaux des systèmes informatiques répartis qui comportent plusieurs ordinateurs reliés entre eux par un système de communication mais localiser géographiquement.

Pour cela, chaque poste de travail peut être branché sur un câble au moyen d'une carte réseau. Le micro-ordinateur peut ainsi travailler indépendamment des autres. Quand il a

besoin d'accéder à l'extérieur, il émet un message sur le réseau définissant le poste ou la ressource adressée. Réciproquement, un message adressé par un autre poste est récupéré par celui ci.

En définitive, chaque carte de réseau surveille la ligne pour recevoir les informations qui lui sont adressées et les décode et, parallèlement elle code les messages émis et les transmet.

La caractéristique principale du réseau local est donc l'autonomie du poste de travail et sa possibilité d'intervenir à tout moment sur le réseau.

Le câble du réseau est donc passif et il n'y a pas d'ordinateur particulier qui serait maître et les autres des esclaves.

Pour permettre la communication et le partage entre postes de travail, il est nécessaire de définir :

- la topologie du réseau ;
- les protocoles de communication ;
- les logiciels de gestion du réseau ;
- les serveurs.

V.3.1.1. LA TOPOLOGIE

On distingue trois formes principales de configuration du réseau local :

- la topologie en bus est la forme la plus souvent rencontrée (figure 19)

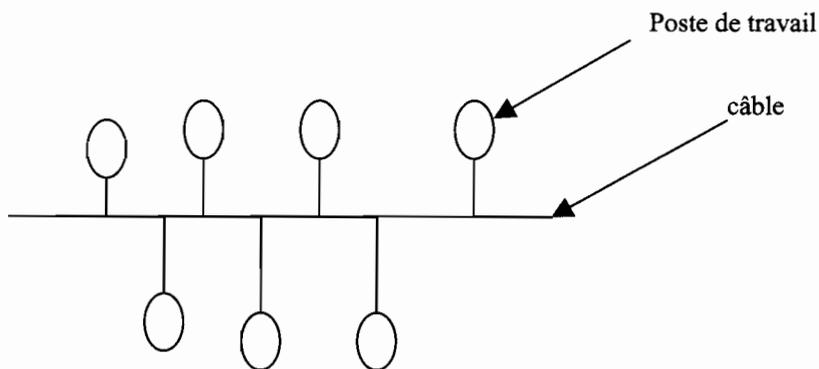


Figure 19 : schéma de représentation d'un réseau en bus.

Dans une topologie en bus, le câble constitue une arête centrale sur laquelle se branche chaque poste de travail. On peut ajouter ou retrancher un élément sans perturber le réseau. Ainsi le câble couvre une distance correspondant à celle séparant les postes les plus éloignés. Dans une telle topologie, l'émission d'un message sur le câble se diffuse sur tout le réseau et donc vers le poste destinataire.

- la topologie en anneau

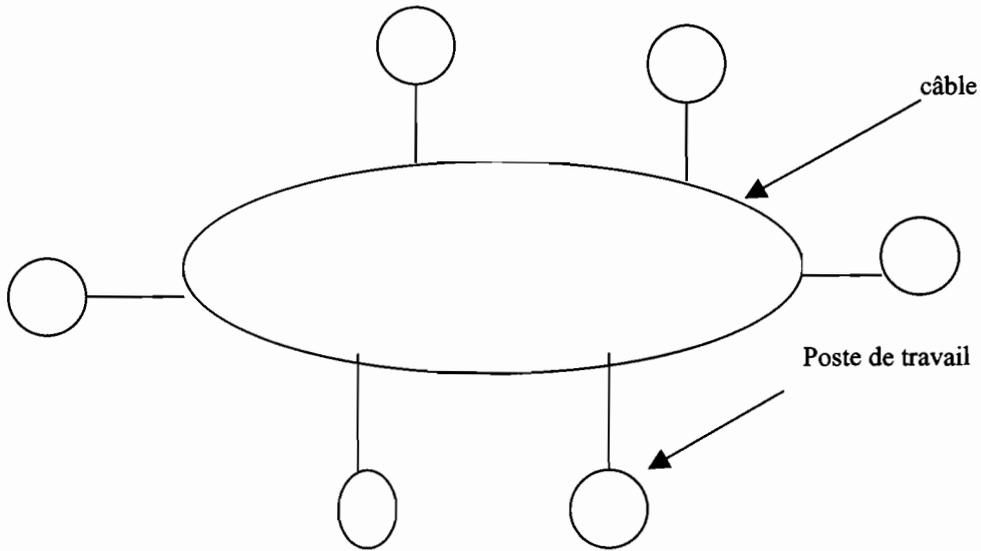


Figure 20 : schéma de représentation d'un réseau en anneau

Un anneau est une boucle fermée reliant tous les postes de travail (figure 20). La différence avec un bus vient de la manière dont les informations émises circulent. Elles tournent sur l'anneau jusqu' à atteindre le poste destinataire. Le sens du parcours est déterminé. Ce qui évite les conflits.

- la topologie en étoile

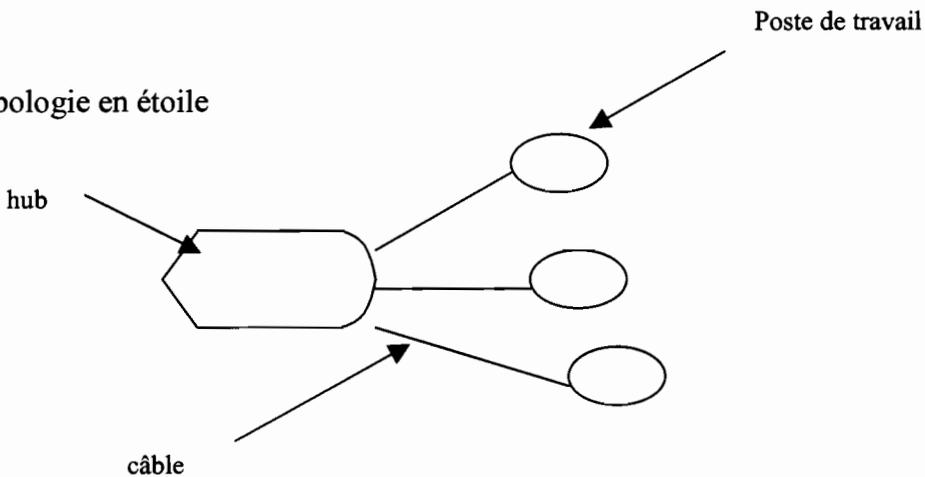


Figure 21 : schéma de représentation d'un réseau en étoile.

Ce n'est qu'une variante des deux topologies précédentes. Elle s'en distingue par le fait que chaque équipement est relié par une connexion à un nœud central appelé Hub ou switch dans le cas d'un réseau ethernet (que nous définirons plus loin). Les performances du réseau vont alors dépendre principalement de ce nœud central.

Un HUB (Host Unit Broadcast) est un nœud de communication utilisé dans les réseau Ethernet 10 base T. Il permet la diffusion. Si un poste de travail envoie un message à

un autre poste, l'information suit le parcours : elle part du poste émetteur, arrive au hub puis est diffusée sur tous les ports du hub ; seul le poste à qui est destiné le message récupère les informations.

Un switch est aussi un nœud de commutation. Lorsqu'un poste veut communiquer avec un autre, l'information arrive au switch qui la transmet uniquement sur le port où est relié le poste concerné par l'information.

V.3.1.1.1. ETHERNET ET TOKEN RING

Il existe deux types majeurs de réseaux locaux : Ethernet et Token ring. Ces deux types de réseaux sont différents de par la façon dont les communications sont faites sur le réseau au niveau standard de communication.

V.3.1.1.1.1. ETHERNET

Toutes les stations connectées sur un réseau ethernet partagent le même câble de transmission. Les signaux sont transmis en série à toutes les autres stations connectées sur le même réseau. Avant de transmettre, une station « écoute » sur le câble afin de s'assurer qu'aucune autre station est en train d'émettre. Si le câble est déjà utilisé, elle attend qu'il devienne disponible. En revanche s'il est libre, elle commence à transmettre ses données sous forme de paquets. Un paquet, frame en anglais, est un bloc d'information. Chaque bloc est structuré de façon identique. Il contient un entête (header) qui contient certaines informations puis une partie des données à transmettre.

Toutes les stations ont une chance de transmettre leur informations sur le réseau. Ceci signifie qu'après chaque transmission de paquets, une station (disons station X par exemple) laisse sa place à d'autres qui voudraient émettre, ce qui assure une répartition équitable de « droit à la parole » que notre station X a de nouveau le droit d'émettre, elle envoie le paquet suivant, et suite jusqu'à ce que toutes les données qu'elle avait à transmettre soient envoyées.

Un réseau standard Ethernet travaille à une vitesse de 10 mégabits/s. Une version plus récente peut travailler à une vitesse de 100 mégabits /s.

Ethernet est le type de réseau le plus répandu, car facile à mettre en place et supporté par des environnements et applications.

Les stations sont reliées entre elles par un câble (figure 22). Il existe deux types de câbles: les câbles coaxiaux et les câbles RJ45. Si le câble utilisé est de type coaxial, les stations sont alors connectées directement entre elles, d'une carte réseau à une autre. Si le câble est de type RJ45, les stations sont alors reliées entre elles par l'intermédiaire d'un hub qui joue un rôle de multiprise et régénérateur de signaux.

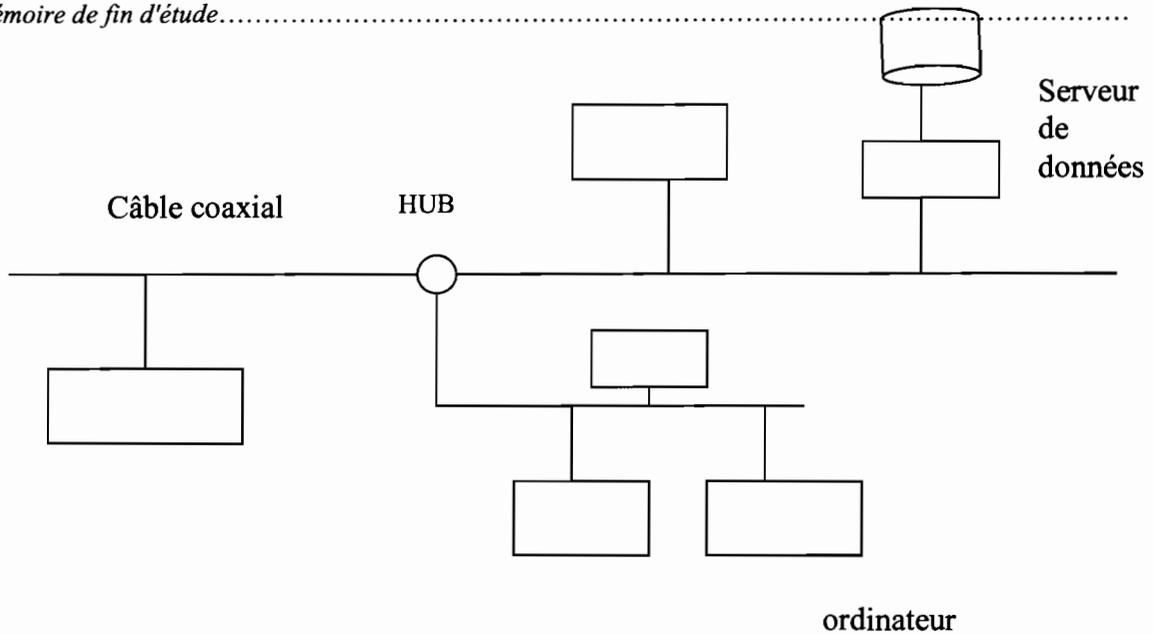


Figure 22 : schéma d'un réseau ethernet

V.3.1.1.1.2. TOKEN RING

Son principe de fonctionnement diffère de celui d'éthernet. Avant de transmettre, chaque station doit s'approprier du jeton lui permettant de le faire. Il n'y a qu'un jeton de communication pour le réseau. Ce qui évite l'émission simultanée de plusieurs trames et résout le problème de l'accès multiples de façon similaire. Les communications sont également effectuées par paquets (virtuellement tous les protocoles de communication en réseau communiquent par paquets) de taille relativement réduite.

Si le transfert est de taille importante, le jeton sera périodiquement distribué afin de permettre à d'autres stations du réseau de faire également leur travail.

Il faut noter que ces deux types de réseaux LAN utilisent les mêmes technologies et offrent presque les mêmes performances. Sur un réseau ethernet, il est possible d'installer une nouvelle station sans pour autant devoir arrêter le réseau. Les stations n'ont pas à attendre le jeton.

De nombreuses études concernant ces deux types de réseaux ont été effectuées. La principale conclusion que l'on peut extraire de ces études est une absence de conclusion qualitative. Ces deux types de réseaux fonctionnent bien et sont pour la plupart du temps suffisamment performants pour bon nombre d'application.

C'est ainsi que les critères de choix ne doivent pas forcément porter exclusivement sur les performances. Nous proposons pour notre système un réseau ethernet connecté à l'Internet.

V.3.1.2. LES PROTOCOLES

A partir du moment où il est possible d'intervenir à tout moment sur le câble réseau, la question primordiale devient celle de s'assurer que chaque message est bien transmis, reçu et décodé. C'est pourquoi, un protocole de communication normalise tout d'abord la manière dont un message est émis et transmis. Lorsqu'un réseau est doté d'un protocole de transfert de

fichiers, il est possible de transférer des fichiers de différents types, et en particulier des fichiers représentant des programmes à exécuter. En fait la soumission d'un travail à distance met en jeu plusieurs fichiers qui peuvent être distribués sur plusieurs sites. Un protocole de soumission de travail à distance permet au site qui demande le travail de communiquer avec celui qui l'exécute. Ce protocole a pour mission de suivre l'évolution du travail en cours, d'informer l'utilisateur de son état d'avancement, et de relancer un transfert de fichier en cas de panne sur un des sites concernés.

Le réseau L.A.N qui sera connecté à l'Internet verra l'installation d'un protocole TCP/IP. Une des bases de ce protocole est d'attribuer une adresse IP propre (comme un numéro de téléphone) à chaque ordinateur du réseau connecté à l'Internet.

V.3.1.3. LES LOGICIELS GESTIONNAIRES DE RESEAUX

Tout ordinateur a besoin d'un système d'exploitation, programme particulier qui permet à l'utilisateur d'interagir avec la machine, de gérer les unités d'entrées et de sortie (clavier, écran, imprimante,...), de démarrer des programmes, de charger et de sauvegarder des fichiers. Il en va de même pour un réseau qui doit gérer tout d'abord les ressources partagées, ensuite les divers utilisateurs autorisés et reconnaître les postes de travail reliés. Une dernière tâche est celle d'assurer la connexion d'une grappe de stations rattachée à un serveur, ce qui suppose donc un adressage de serveur (si les réseaux locaux sont connectés), des stations et des autres matériels (imprimante,...). C'est le rôle d'un logiciel gestionnaire de réseau d'assurer ces divers services. Les plus répandus étant notamment Netware (Novell), Lan Manager (Microsoft) ou OS/2 et Lan Server (IBM).

V.3.1.4. SERVEURS

La ressource la plus importante au sein d'un réseau local est constituée par le serveur. Il s'agit le plus souvent d'un micro-ordinateur puissant tant en mémoire vive, en capacité de disques que de vitesse du processeur. C'est sur ce serveur que se trouve installer le logiciel de gestionnaire réseau. La notion de serveur est alors représentée pour l'utilisateur, par les services offerts au niveau du stockage des logiciels auquel chaque utilisateur aura recours ainsi que des fichiers que cet utilisateur pourra sauvegarder ou charger.

A côté de ces serveurs, on trouve aussi des serveurs plus spécialisés et adaptés à des services particuliers tels que serveurs de bases de données, d'imprimantes, etc.

VI. DETERMINATION DU COUT DU SYSTEME

L'estimation des coûts est une étape importante dans le développement d'un logiciel ou d'un système. Elle permet à l'utilisateur de prendre connaissance du coût du matériel, du coût de développement.

Cette étape qui permet à l'ingénieur de faire le choix du matériel par rapport au budget de l'utilisateur sera composée de l'estimation du coût de développement et de celui du matériel.

VI.1. ESTIMATION DU COUT DE DEVELOPPEMENT

La majeure partie des coûts de développement correspond aux frais salariaux des ingénieurs ou des programmeurs logiciels. Le coût d'un projet est directement proportionnel au nombre d'homme mois nécessaires à la réalisation du projet. Le coût d'un projet est généralement basé sur l'estimation de la taille du code source, de l'expérience des programmeurs, de la complexité du domaine, les contraintes de réalisation.

Nous nous limiterons à la description de la méthode de Jones et d'un modèle d'estimation générale le COCOMO.

VI.1.1. ESTIMATION PAR LA METHODE DE JONES

Plutôt que d'estimer les coûts en utilisant un historique de productivité en nombre de lignes sources par programmeur et par mois, Jones suggère un calcul des coûts basé sur une unité de coût correspondant à la production d'un milliers de lignes sources. La valeur de cette unité de coût est obtenue en divisant le coût total d'un projet passé par la taille du programme en KLS (Kilo Ligne Source).

Un des avantages de cette méthode est que tous les coûts associés au projet peuvent être pris en compte uniformément.

Pour l'estimation des coûts, Jones a introduit la notion de rectangle de probabilité. Cette technique repose toujours sur une estimation de la taille du programme. Elle s'appuie sur des historiques pour déterminer les tailles minimale et maximale du programme. De plus, une estimation minimale et maximale du coût de développement par KLS est donnée. On obtient ainsi un rectangle de probabilité comme indiqué par la figure 23 ci dessus.

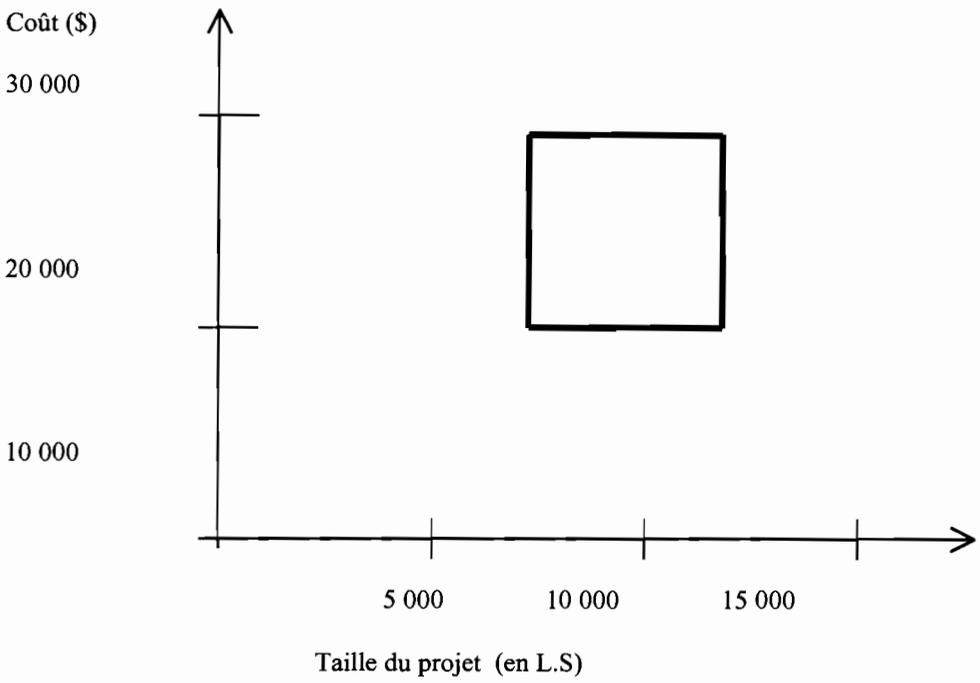


Figure 23 : schéma d'estimation par la méthode JONES

Le coût minimum de développement est de (8 KLS * 20 000) soit 160 000 \$ alors que le coût maximum est de (11 KLS * 28 000) soit 308 000 \$.

VI.1.2. ESTIMATION PAR LA METHODE COCOMO

Le modèle le mieux documenté dont les paramètres sont adaptables à l'environnement est le modèle COCOMO. COCOMO est un acronyme pour Constructive Cost Model. C'est une méthode d'estimation du coût d'un projet logiciel dans le but d'éviter les erreurs de budget et les retards de livraison, qui sont malheureusement habituels dans l'industrie de développement des logiciels. Ce modèle existe en trois versions: simple ou modèle de base, intermédiaire et détaillé. Nous présenterons seulement les grandes lignes du modèle de base.

Le modèle de base estime l'effort (en nombre d'homme-mois) en fonction du nombre de lignes de code, la productivité (le nombre de lignes de code par personne par mois) et un facteur d'échelle qui dépend du type de projet. Les trois types de projets identifiés sont :

- Organique : ces projets sont réalisés par une équipe de taille relativement petite travaillant dans un environnement familier et dans un domaine d'application connu de l'équipe. C'est par exemple le cas d'un système de notes dans une école.
- Embarqué : Le système à développer est une partie d'un système complexe alliant fortement des matériels, des logiciels, des normes et des procédures opérationnelles. C'est par exemple le cas des outils de C.AO.
- Semi-détaché : ce mode représente un intermédiaire entre les deux précédents. Pour des projets de ce type, l'équipe peut être composée de programmeurs de divers niveaux d'expérience. Les membres peuvent être totalement inexpérimentés en ce qui concerne quelques uns des aspects du système à développer, mais pas tous. C'est par exemple le cas de notre système de gestion des enseignements.

Les formules permettant de calculer le coût, ou plus exactement l'effort requis pour le développement du logiciel sont les suivantes :

- Mode organique $HM = 2,4 * (KLS)^{1,05}$
- Mode semi-détaché $HM = 3 * (KLS)^{1,12}$
- Mode embarqué $HM = 3,6 * (KLS)^{1,20}$

Le modèle COCOMO permet également d'estimer le temps de développement nécessaire au projet (TDEV). Le temps de développement est le temps requis pour terminer le projet, en supposant que les ressources de personnel requises sont disponibles. Les équations pour les différents modes de projets sont les suivantes :

- Mode organique $TDEV = 2,5 * (KLS)^{0,38}$
- Mode semi-détaché $TDEV = 2,5 * (KLS)^{0,35}$
- Mode embarqué $TDEV = 2,5 * (KLS)^{0,32}$

La variation des résultats obtenus ne discrédite pas l'utilisation de la modélisation en elle-même. Elle illustre simplement le fait que les paramètres associés à chaque modèle dépendent très largement du contexte d'utilisation de la méthode. A cause des techniques différentes utilisées pour mesurer des quantités telles que le nombre de lignes de code, le nombre de jours de travail par personne et par mois, etc. , il n'est pas possible d'établir un modèle d'évaluation des coûts qui soit universel.

Nous utiliserons la méthode COCOMO qui reste la référence en matière d'estimation détaillée des coûts et surtout de la ventilation de ces coûts suivant les phases des projets. Les estimations de COCOMO sont d'autant plus fiables que les paramètres du projet sont bien connus.

VI.1.3. ESTIMATION DU COUT DE NOTRE SYSTEME

Le nombre de KLS sera le nombre de fonction de l'architecture du moteur d'inférence

$$HM = 3 * (14)^{1,12} = 57$$

$$TDEV = 2,5 * (57)^{0,35} = 10 \text{ Mois}$$

En prenant le salaire minimum d'un ingénieur de conception étant de 225 000 F CFA, nous pouvons estimer le coût de développement $10 * 225\ 000 \text{ F} = 2\ 225\ 000 \text{ F CFA}$.

VI.2. ESTIMATION DU COUT DU MATERIEL

Presque tous les matériels (ordinateurs, serveur, hub, ...) pour la mise en œuvre du système. Mais il faudra une ligne spécialisée pour la connexion Internet et des câbles réseau pour la mise en œuvre du réseau local. Le tableau 4 propose des prix extraits du site <http://reseaux.info.free.fr/français/topologie.htm> le 18 décembre 2001.

Désignation	Caractéristiques	Prix unitaire (CFA)	Nombre	Coût total (CFA)
Câble réseau	Câble coaxial	300 F le mètre	1000 m	300 000 F
Ligne Spécialisée				1 200 000 F
TOTAL				1 500 000 F

Les frais d'installation sont estimés à 800 000 Fcfa (information recueillie auprès de l'informatique service)

Tableau 4 : tableau récapitulatif du coût du matériel.

VII. REALISATION

Par manque d'outils proposés pour l'implémentation de cette solution, nous avons utilisé Visual Basic 6 pour présenter quelques parties de l'application de gestion des enseignements.

- La page d'accueil : elle est constituée de l'image de l'U.P.B et permet à l'utilisateur d'avoir le menu de réalisation des tâches. Elle est représentée par l'interface de la figure 24 suivante :



Figure 24 : interface proposé pour la page d'accueil de l'application.

- Le menu : également constitué de l'image de l'U.P.B, il permet de présenter les tâches réalisables par le système. Il présente les fonctions demandées par l'utilisateur soit en accédant directement, soit en passant par un sous menu. L'interface de la figure 25 est une présentation du menu de travail.



Figure 25 : interface du menu de l'application.

- le sous menu de la génération : il permet de générer la fiche programmatique d'un enseignant pour une période donnée. Il permet également de proposer l'emploi du temps d'une semaine. Il permet de produire les états périodiques d'un niveau d'enseignements et la fiche financière d'un enseignant au cours d'une période (figure 26).



Figure 26 : interface du sous menu génération

- l'écran de génération d'un emploi du temps : cet écran permet de saisir l'établissement, le niveau, le cycle, la période pour laquelle on souhaite avoir une proposition d'emploi du temps.

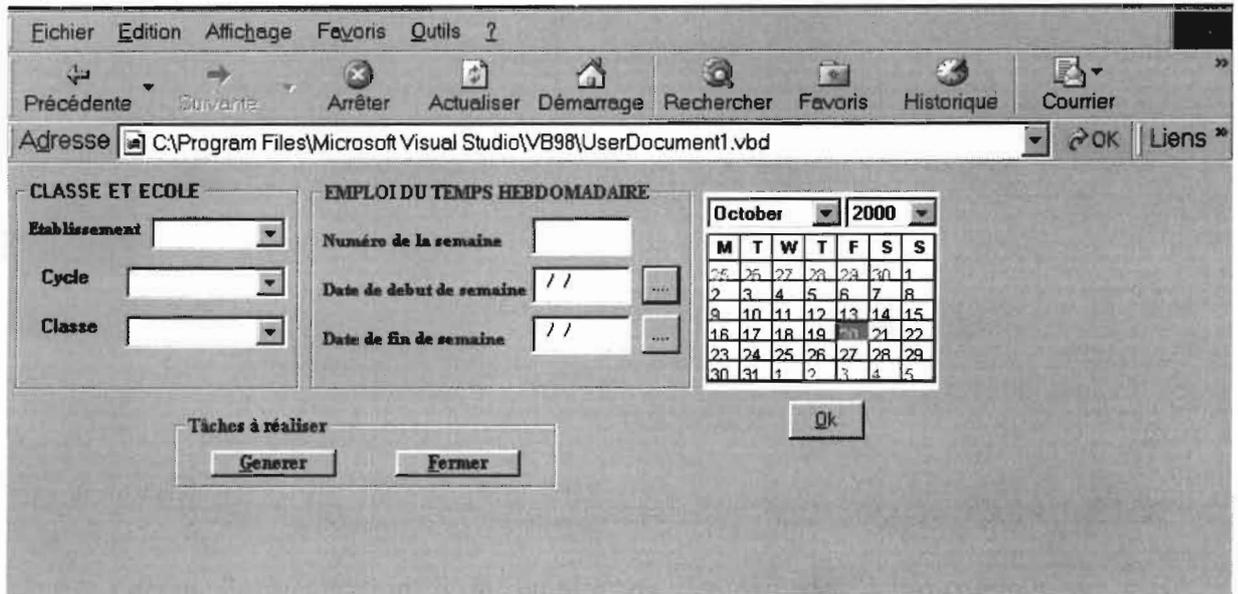


Figure 27 : interface de génération d'une proposition d'emploi du temps hebdomadaire.

- écran de mise au point d'un emploi du temps : cet écran récupère l'emploi du temps proposé par le moteur d'inférence. Il permet à l'utilisateur de faire des modifications et de contrôler leur validité.

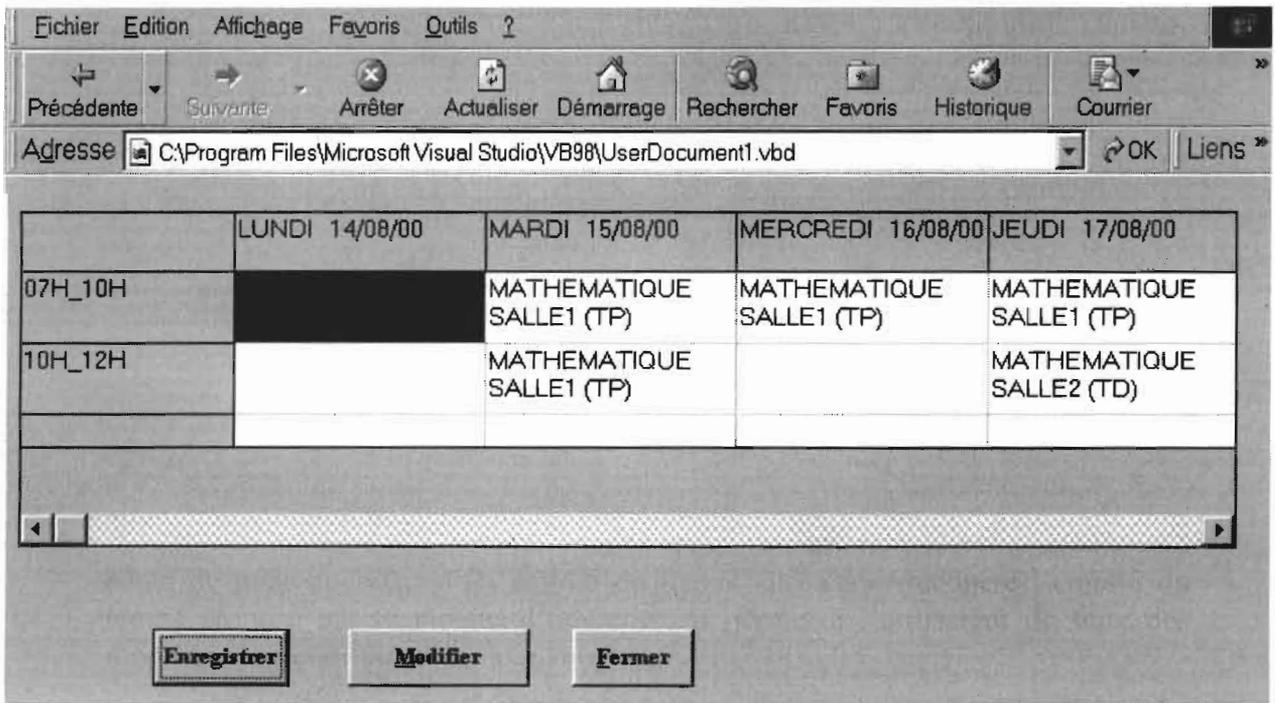


Figure 28 : écran de mise au point d'un emploi du temps.

CONCLUSION

Ce document résume les bases de la mise en œuvre d'un système de " gestion des enseignements" plus particulièrement le cas de l'ESI. Il décrit le système voulu par le demandeur et les étapes nécessaires pour sa réalisation.

Le thème de cette étude "gestion des enseignements " qui est seulement consacré ici à l'établissement de la fiche programmatique, la réalisation des emplois du temps hebdomadaires en fonction des contraintes, l'établissement des fiches financières et d'évaluation des cours pourrait être étendu à d'autres aspects comme :

- la gestion des notes des étudiants pour:
 - une information en temps réel des étudiants par rapport à leurs notes ;
 - un calcul rapide des moyennes des élèves ;
 - une meilleure conservation de ces notes ;
- la gestion de la bibliothèque pour permettre aux enseignants et aux étudiants de connaître les documents de la bibliothèque et un meilleur suivi de ces documents ;
- le transfert par Intranet des fiches financières des enseignants pour le service financier du rectorat après validation par le chef de département, par le directeur des études et par le directeur et après prise de connaissance par les intéressés.

Outre les suggestions d'étendre le système, la tâche qui nous reste est de reprogrammer cette application avec les outils demandés.

BIBLIOGRAPHIE

OUVRAGES

1. [CORNAFION 1981] CORNAFION, **Systèmes informatiques Répartis – concepts et techniques**, Edition DUNOD, 1981.
2. [GARDARIN 1994] Georges GARDARIN, **base de données. Les systèmes et leurs langages**, Edition Berti Editions, 1995.
3. [G.GARDARIN ET P.VALDURIEZ 1991], G.GARDARIN , P.VALDURIEZ, **SGBD AVANCES. Bases de données objets, déductives, réparties**, Edition Eyrolles, 1991.
4. [MOKRANE BOUZEGHOUB, GEORGES GARDARIN,PATRICK VALDURIEZ 1995]
5. [Georges GARDARIN, Mokrane BOUZEGHOUB,Patrick VALDURIEZ], **Du C++ à Merise Objet. Concepts, Langages, Bases de données, Méthodes, Interfaces**, Edition Eyrolles, 1995.
6. [IGOR ALEKSANDER], Igor ALEKSANDER, **La conception des systèmes intelligents**, Edition HERMES.
7. [JEAN-CHARLES POMEROL] Jean-CHARLES POMEROL, **Les systèmes experts**, Edition HERMES.
8. [IAN SOMMERVILLE] Ian SOMMERVILLE, **Le génie logiciel et ses applications**, Edition InterEditions.
9. [JEAN-CLAUDE COURBON] Jean-Claude COURBON, **SYSTEMES D'INFORMATION : STRUCTURATION, MODELISATION ET COMMUNICATION**, Edition InterEditions.

SITES CONSULTES

- <http://www.essi.fr/~hugues/GL/COCOMO/cocomo.html>
- <http://www.ifrance.com/mondelphi/comparevbdelphi.htm>
- <http://www.papyrusweb.ch/cours/win9xNetwork.htm>
- <http://reseaux.info.free.fr/français/topologie.htm>
- <http://www.chez.com/jeuxreseau/reseau1.html>

SCHEMAS UTILISES

Tableau 1 : récapitulatif des informations du M.C.D.....	11
Tableau 2 : explication de la structure du moteur d'inférence.....	27
Tableau 3 : Tableau comparatif de VB et de DELPHI.....	37
Tableau 4 : tableau récapitulatif du coût du matériel.	45
Figure 1 : illustration des concepts du modèle entité-association.....	8
Figure 2 : illustration des concepts de méthodes fonctionnelles.....	9
Figure 3 : : le modèle conceptuel de données	10
Figure 4 : scénario d'établissement de la fiche programmatique.....	13
Figure 5 : scénario d'établissement de la fiche financière.....	14
Figure 6 : scénario de proposition d'emploi du temps hebdomadaire.....	15
Figure 7 : schéma d'une approche de base de données distante	17
Figure 8 : schéma d'une approche de base de données répartie	18
Figure 9 : schéma de l'architecture générale de notre système.....	19
Figure 10 : Base homogène (division d'une base de données en bases de données locales)..	20
Figure 11 : Base hétérogène (intégration des bases de données existantes en une base de données).....	20
Figure 12 : schéma d'explication d'un SGBD.....	22
Figure 13 : architecture du système de gestion des enseignements	23
Figure 14 : le fonctionnement du moteur d'inférence	25
Figure 15 : structure du moteur d'inférence	27
Figure 16 : schéma d'explication de la règle 1 de passage d'une entité-association à un modèle relationnel	32
Figure 17 : schéma d'explication de la règle 2 de passage d'une entité-association à un modèle relationnel	32
Figure 18 : schéma d'explication de la règle 3 de passage d'une entité-association à un modèle relationnel	33
Figure 19 : schéma de représentation d'un réseau en bus.....	38
Figure 20 : schéma de représentation d'un réseau en anneau.....	39
Figure 21 : schéma de représentation d'un réseau en étoile.....	39
Figure 22 : schéma d'un réseau ethernet	41
Figure 23 : schéma d'estimation par la méthode JONES	43
Figure 24 : interface proposé pour la page d'accueil de l'application.	46
Figure 25 : interface du menu de l'application.....	47
Figure 26 : interface du sous menu génération.....	48
Figure 27 : interface de génération d'une proposition d'emploi du temps hebdomadaire....	49
Figure 28 : écran de mise au point d'un emploi du temps.	50

GLOSSAIRE

- A : utilisé pour définir les types de données au niveau du modèle conceptuel de données. Cette lettre signifie alphabet.
- AN : utilisé pour définir les types de données au niveau du modèle conceptuel de données. Ce mot signifie Alpha Numérique.
- N : utilisé pour définir les types de données numériques au niveau du modèle de données.
- BAS : c'est l'extension des fichiers globaux sous Visual Basic.
- CAO: Conception Assistée par Ordinateur.
- D : utilisé pour définir les types de données au niveau du modèle conceptuel de données. Cette lettre signifie Date.
- DHTML: Dynamic Hyper Text Markup Language
- DLL: Dynamic Language Library
- DCU: fichier d'unité de Pascal
- Ethernet : Réseau local se présentant sous plusieurs formes (étoile ou bus) et à différents débits. C'est aussi un protocole de liaison de données qui fonctionne dans les couches physique et liaison du modèle OSI (couche 1 et 2).
- HUB : Host Unit Broadcast (concentrateur). C'est un nœud de communication utilisé dans les réseaux Ethernet 10 base T (topologie en étoile). Un HUB diffuse les données sur le réseau.
- HTML : Hyper Text Markup Language.
- Intranet : Réseau offrant un jeu de protocoles connexes et de fichiers Internet standard au format HTML à des utilisateurs.
- ISO : International Standards Organisation. Organisme international qui élabore les standards comme OSI.
- IBM : International Business Machine.
- L4G : Langage de 4^{ème} Génération.
- LAN : Local Area Network. Réseau locaux (à l'échelle d'un immeuble).
- MAU : Multistation Access Unit. C'est un nœud de communication utilisé dans les réseaux types Token – ring mais avec une topologie physique en étoile. La topologie logique reste en anneau. Les MAU évitent les ruptures de l'anneau quand une station se déconnecte.
- OCX: fichier de commandes en visual basic.
- OSI : Open Systems Interconnection. Il s'agit d'un modèle de référence pour les protocoles. L'OSI est défini sur 7 couches (physique, Liaison, Réseau, Transport, Session, Présentation, Application).
- Protocole : ensemble de règles définissant un langage de communication.
- SQL: Structured Query Language.
- Webclasses : un objet webclass est un composant Visual Basic résidant sur un serveur Web et répondant aux entrées du navigateur.
- HM : homme mois.
- TDEV : temps de développement.
- Topologie d'un réseau : Relation physique et logique entre les nœuds. Les plus courantes sont en étoile, en bus, en anneau ou une combinaison de ces précédents types.
- KLS kilo Ligne du code Source.

- Switch : (commutateur). Comme le HUB, il s'agit d'un nœud de communication. Un switch possède une table d'adresses des cartes réseaux connectées sur ports. Les données sont ainsi envoyées uniquement au(x) destinataire(s).
- TCP/IP : ce terme désigne à la fois un protocole et une architecture de protocole (basé sur 4 couche).
- TCP : signifie Transmission Control Protocol. C'est un protocole qui vérifie le bon fonctionnement d'un paquet IP. Elle fournit aussi d'autres service : l'établissement d'une liaison, le séquençage des paquets, le contrôle de flux, la gestion des erreurs, le message d'établissement d'une liaison.
- IP : signifie Internet Protocol. : c'est un protocole qui ne s'occupe que de l'acheminement des paquets IP.