

Département Génie Informatique

BD40

Christian FISCHER

Copyright © Septembre 2002

SOMMAIRE

MODULE 1	MAQUETTAGE D'APPLICATIONS INFORMATIQUES ET PROGRAMMATION EVENEMENTIELLE.....	5
1. PRÉSENTATION	5	
1.A. REFERENCES TECHNIQUES.....	5	
1.B. REFERENCES BIBLIOGRAPHIQUES.....	6	
2. DÉVELOPPEMENT D'APPLICATIONS EN ENVIRONNEMENT GRAPHIQUE.....	7	
2.A. INTRODUCTION.....	7	
2.B. L'INTERFACE HOMME-MACHINE EN ENVIRONNEMENT GRAPHIQUE.....	7	
2.C. CRITERES ERGONOMIQUES	9	
2.D. LA STYLIQUE DES APPLICATIONS GRAPHIQUES.....	11	
2.E. LE CONCEPT WIMP	12	
2.F. LA NORME CUA.....	14	
2.G. LA CONCEPTION DES INTERFACES	15	
2.H. LE MAQUETTAGE.....	17	
3. LES NORMES DE PRÉSENTATION	19	
3.A. DES APPLICATIONS WINDOWS.....	19	
3.B. DES APPLICATIONS INTERNET ET INTRANET	32	
4. LA PROGRAMMATION ÉVÉNEMENTIELLE.....	41	
4.A. QUELQUES PRINCIPES	41	
4.B. LES DIFFERENTS TYPES D'EVENEMENTS.....	43	
4.C. LES OBJETS GRAPHIQUES.....	43	
5. INTERFACE GRAPHIQUE ET PROGRAMMATION EVENEMENTIELLE AVEC ACCESS ET VISUAL BASIC	45	
5.A. ORGANISATION DE L'INTERFACE	46	
5.B. PROGRAMMATION EVENEMENTIELLE DE L'INTERFACE	47	
MODULE 2	SYSTEME DE GESTION DE BASE DE DONNEES RELATIONNEL.....	51
1. PRÉSENTATION	51	
1.A. REFERENCES TECHNIQUES.....	51	
1.B. REFERENCES BIBLIOGRAPHIQUES.....	52	
2. NOTION DE BASE DE DONNÉES.....	53	
2.A. QU'EST-CE QU'UN SGBDR ?.....	53	
2.B. CARACTERISTIQUES.....	53	
3. ÉTAT DU MARCHÉ DES SGBDR	55	
3.A. LES GENERATIONS DE SGBD	55	
3.B. LES SGBD RELATIONNELS.....	56	
4. UN SGBDR : MICROSOFT ACCESS.....	59	
4.A. LE STOCKAGE ET L'ACCES AUX DONNEES.....	59	
4.B. LE SCHEMA PHYSIQUE D'UNE BASE DE DONNEES	60	
4.C. INTEGRITE D'UNE BASE DE DONNEES.....	67	
4.D. SECURITE D'UNE BASE DE DONNEES	74	
4.E. REQUETES	76	
4.F. LES OUTILS DE DEVELOPPEMENT	79	
5. LE LANGAGE SQL	84	
5.A. LE STANDARD SQL	84	
5.B. L'ALGÈBRE RELATIONNELLE.....	85	
5.C. SQL: UN LANGAGE DE MANIPULATION DES DONNEES.....	94	

SOMMAIRE

MODULE 3 METHODES DE CONCEPTION DE SYSTEMES D'INFORMATIONS110

1. PRÉSENTATION	110
1.A. REFERENCES TECHNIQUES.....	110
1.B. REFERENCES BIBLIOGRAPHIQUES.....	110
2. RÉALISATION D'UN PROJET INFORMATIQUE	111
2.A. QU'EST-CE QU'UNE ANALYSE INFORMATIQUE ?	111
2.B. SYSTEME D'INFORMATION (SI)	111
2.C. METHODES D'ANALYSE.....	112
2.D. LA DÉMARCHE DE CONCEPTION	113
2.E. LE DÉVELOPPEMENT	113
2.F. LES TESTS DE LOGICIEL	114
3. LA MÉTHODE D'ANALYSE MERISE	115
3.A. PRESENTATION	115
3.B. CARACTERISTIQUES.....	115
3.C. LES MODELES DE MERISE.....	116
3.D. LE SUIVI D'UN PROJET MERISE	117

MODULE 4 GENIE LOGICIEL : METHODES ET OUTILS123

1. PRÉSENTATION	123
1.A. REFERENCES TECHNIQUES.....	123
1.B. REFERENCES BIBLIOGRAPHIQUES.....	123
2. LES CONCEPTS DU GÉNIE LOGICIEL	124
2.A. L'ATELIER DE GENIE LOGICIEL (AGL).....	124
2.B. ÉVOLUTION DES ATELIERS DE GENIE LOGICIEL	125
2.C. LES FONCTIONS ET LES COMPOSANTS D'UN AGL	126
2.D. LES FONCTIONS ET LES COMPOSANTS D'UN AGL	127
3. LE MARCHÉ DU GÉNIE LOGICIEL	129
3.A. LES ATELIERS DE GENIE LOGICIEL (AGL)	129
3.B. LES OUTILS DE GENIE LOGICIEL (OGL).....	131
3.C. LES METHODES UTILISEES	132
3.D. LES OUTILS UTILISES	133
3.E. LA METHODE MERISE ET LES OUTILS ASSOCIES	134
3.F. MERISE ET AMC*DESIGNOR	135
4. LES ÉTAPES D'UN PROJET INFORMATIQUE.....	138
4.A. DECOUPER LA METHODOLOGIE DE DEVELOPPEMENT EN ETAPES	138
4.B. PARCOURIR LES ETAPES DE GESTION DU PROJET LOGICIEL.....	138
4.C. LES MODELES DE CYCLE DE VIE	142

MODULE 5 MODELISATION DES COMMUNICATIONS145

1. PRÉSENTATION	145
1.A. REFERENCES TECHNIQUES.....	145
1.B. REFERENCES BIBLIOGRAPHIQUES.....	145
2. MODÉLISATION DES COMMUNICATIONS.....	146
2.A. OBJECTIFS.....	146
2.B. VOCABULAIRE	146
2.C. METHODE	149

SOMMAIRE

MODULE 6	MODELISATION DES TRAITEMENTS.....	150
1. PRÉSENTATION		150
1.A. REFERENCES TECHNIQUES.....		151
1.B. REFERENCES BIBLIOGRAPHIQUES.....		151
2. MODELISATION CONCEPTUELLE DES TRAITEMENTS		152
2.A. IDENTIFICATION DES OPERATIONS CONCEPTUELLES.....		152
2.B. REPRESENTATION GRAPHIQUE D'UNE OPERATION CONCEPTUELLE		152
2.C. EXEMPLE DE MCT		153
2.D. REGLES DE CONSTRUCTION D'UN MCT		153
2.E. VALIDATION		155
2.F. CYCLES		156
2.G. DELAIS.....		156
3. MODELISATION ORGANISATIONNELLE DES TRAITEMENTS		158
3.A. LES PROCEDURES FONCTIONNELLES		158
3.B. VALIDATION DONNEES - TRAITEMENTS.....		160
4. ARCHITECTURE TECHNIQUE.....		162
4.A. DEFINITION.....		162
4.B. CONCEPTION DUNE ARCHITECTURE TECHNIQUE AVEC TACT		164
4.C. EXEMPLE D'ARCHITECTURE TECHNIQUE.....		167

Module 1 Maquettage d'applications informatiques et programmation événementielle

1. PRÉSENTATION

Cette séquence présente la notion d'interface homme-machine, les critères ergonomiques ainsi que des normes de présentation rassemblés dans un guide de style pour vous permettre de développer des applications graphiques.

Ces notions font partie des « fondamentaux » de la culture du technicien supérieur en informatique de gestion. Elles visent à vous donner une culture de base portant notamment sur les normes et standards de fait en matière de présentation et d'ergonomie d'interfaces graphiques

La programmation événementielle sera abordée comme le mode habituel de développement d'applications en mode graphique comportant l'utilisation de procédures et d'objets dédiés à l'environnement.

Les fonctionnalités des outils de création d'applications multimédia (hors ligne comme les CD-ROM ou en ligne comme les pages des sites WEB) seront montrées sur des exemples concrets (organisation des répertoires et fichiers d'un CD-ROM, structure et codage d'une page HTML).

1.A. Références techniques

Le logiciel Microsoft Access et le langage Visual Basic pour application sont utilisés pour illustrer la création et la programmation des interfaces graphiques.

Avec Access, la programmation se fait à l'aide du langage Visual Basic pour Applications (VBA). Ce langage est commun à toutes les applications Microsoft de la gamme Office : Word, Excel, Access notamment.

Dans les versions précédentes de Access (version 1 et 2) le langage Visual Basic était appelé Access Basic.

L'environnement de développement Visual Basic pour Windows utilise lui aussi, bien entendu, ce langage VBA. La construction de l'interface graphique et sa programmation suit la même logique dans les environnements de développement Access et Visual Basic.

Parmi les outils de génération d'applications, nous préconisons l'utilisation d'AMC*Designor ou WinDesign qui offrent la possibilité de générer une maquette avec une interface graphique pour Visual Basic.

1.B. Références bibliographiques

Parmi les ouvrages nous recommandons notamment :

- Interface graphique : Interface Windows - Guide d'ergonomie chez Microsoft Press
- Guide de style pour les applications Windows - SQL Ingénierie, 1995
- L'interface utilisateur, pour une informatique plus conviviale - Meinadier, Dunod, 1991
- The Windows Interface : An application design guide - Microsoft Windows Software Development Kit
- L'interface graphique ; enjeux, ergonomie, mise en oeuvre sous Windows Gillet, InterEdition, 1995

Parmi les ressources disponibles sur Internet, vous pouvez consulter

- Le site Internet de Sql Ingénierie : <http://www.sqli.fr>

Ce site français est une mine d'informations techniques et stratégiques. Sql Ingénierie propose le téléchargement de tutoriels, des études comparatives d'outils de développement et de nombreux documents (téléchargeables au format Acrobat), dont des extraits du "Guide de style pour Windows 95" et du "Guide d'ergonomie des applications Intranet". De plus, sur ce site il est question du marché des outils client-serveur, du 2^{ème} âge du client-serveur, du client-serveur universel sous forme de FAQ. Sql Ingénierie fournit par ailleurs le cahier des charges de l'étude comparative sur les outils client-serveur pour les éditeurs.

- Le site Internet de WebDesigner : <http://web.canlink.com/webdesign>

Ce site présente un guide complet de la conception d'applications Internet : description et mise en œuvre de HTML, réalisation d'une page Web, les dix commandements de HTML, etc.

2. DÉVELOPPEMENT D'APPLICATIONS EN ENVIRONNEMENT GRAPHIQUE

2.A. Introduction

L'évolution des technologies informatiques a permis la mise au point de nouveaux types d'applications à destination de nouveaux utilisateurs.

Aux débuts de l'informatique, les ordinateurs coûtent chers donc il est plus avantageux de faire supporter l'effort d'adaptation à l'homme plutôt qu'aux systèmes informatiques. Par ailleurs les technologies du moment ne permettaient pas de faire autrement.

Aujourd'hui, les avancées dans le domaine des techniques logicielles et matérielles permettent d'offrir de nouveaux modes d'interaction entre l'homme et la machine : l'interface graphique remplace le langage de commande. Qui plus est, la diffusion des ordinateurs personnels permet de familiariser le public avec l'outil.

Le coût du rejet d'un système informatique par l'utilisateur du fait de son inadéquation au besoin est devenu un enjeu important.

Pour construire une application atteignant les critères de qualité en terme *d'utilisabilité* de l'interface, les informaticiens doivent comprendre l'activité et la façon de travailler de l'utilisateur afin que l'outil informatique ne soit pas une gêne pour lui mais, bien au contraire, lui permette de se concentrer sur sa tâche.

Or, pour effectuer cette prise en compte du facteur humain, les informaticiens ne sont guère outillés :

- les méthodes sont axées sur le développement fonctionnel de l'application et non pas sur l'utilisateur et son interaction avec le système ;
- en termes d'analyse de l'existant et d'expression des besoins, la pratique est basée sur l'analyse des procédures de l'organisation et des flux de données mais pas ou peu sur le travail de l'utilisateur final.

2.B. L'INTERFACE HOMME-MACHINE EN ENVIRONNEMENT GRAPHIQUE

Une **interface homme-machine** est un ensemble de règles gérant un espace de communication et permettant à deux acteurs par nature différents, l'homme et la machine, d'échanger des informations.

Une interface homme-machine est un ensemble de moyens informatiques (logiciels et matériels) permettant au couple (machine, utilisateur) de « communiquer » en vue de l'accomplissement d'une tâche requise par l'utilisateur de la manière la plus conviviale, agréable, facile, simple possible.

L'interface homme-machine a d'abord existé en mode caractères (à partir de 1975), puis en mode graphique (à partir de 1985) et devient aujourd'hui multimédia (intègre images, sons, vidéos).

SOMMAIRE

Les **interfaces graphiques** (GUI - *Graphical User Interface*) sont fondées sur le concept de **métaphore**.

Leur objectif est de procurer à l'utilisateur des objets porteurs de sens, ne nécessitant pas la mémorisation du concept sous-jacent.

L'idée métaphorique des objets d'une interface graphique est celle du bureau de travail : les objets dont nous disposons sur notre bureau nous sont familiers ; en jetant un simple coup d'oeil, nous savons à quoi ils servent.

Les objets ainsi que les opérations qui leur sont associés constituent le fondement des interfaces graphiques.

Avec les notions de multifenêtrage et d'icônes, elles permettent de mettre en place un dialogue dit **à plusieurs fils d'activité**.

Par rapport aux interfaces en mode caractères, une interface graphique offre :

- plus d'informations,
- un travail multi-applications voire multi-sessions,
- de nouveaux périphériques d'entrée,
- une interaction pilotée par l'utilisateur,
- des manipulations plus faciles
- un apprentissage plus rapide
- un meilleur confort d'utilisation

mais nécessite plus de ressources (processeur, mémoire, disque).

L'interface graphique n'est pas seulement utile pour les tâches de bureautique, son bénéfice est également important pour toutes les applications de gestion réclamant beaucoup de saisie.

L'utilisation du mode graphique doit permettre :

- d'une part, une meilleure productivité de l'utilisateur (plus d'automatisation, moins d'erreurs, meilleure appropriation) et **d'aller plus vite** en faisant moins d'erreurs de frappe et de fautes d'orthographe qu'avec de classiques applications en mode texte ;
- d'autre part, de faciliter la formation grâce à une interface plus intuitive et à des éléments communs à plusieurs applications.

2.C. Critères ergonomiques

L'objectif du concepteur de l'interface est de produire une représentation perceptible la plus proche possible de la représentation mentale de l'utilisateur.

Prise en compte de la charge de travail

La charge de travail se définit comme la sollicitation des capacités de perception, de mémorisation, de traitement de l'information et d'apprentissage de l'utilisateur pour effectuer son travail.

L'interface doit prendre en compte la charge de travail de l'utilisateur en lui permettant :

- d'optimiser la prise de décision. Il s'agit de présenter l'information la plus brève et la moins ambiguë possible,
- de minimiser le nombre d'actions ou d'opérations à effectuer ainsi que le temps de manipulation et
- d'éviter la surcharge ou la sous-charge de travail.

Adaptabilité et flexibilité

Une interface est dite adaptable ou flexible lorsque elle utilisable par des types d'utilisateurs différents.

Une telle interface doit donc être construite de manière à ce que l'utilisateur novice puisse réaliser les tâches les plus courantes et que les utilisateurs les plus expérimentés puissent améliorer leur productivité.

Transparence

Les modes d'implantation des objets de l'application et leur lien ne doivent pas être visibles pour l'utilisateur. Seul le déroulement des actions sur les objets est visible.

Facilité d'apprentissage

L'apprentissage de l'interface doit pouvoir être empirique : l'utilisateur apprend en utilisant un système navigationnel. Il découvre ainsi les fonctionnalités de son application, tout en travaillant sur sa tâche.

Pour obtenir cette aisance, il est nécessaire :

- de présenter une information concise non superflues,
- de réduire la charge informationnelle en ne présentant à l'écran que les informations utiles à un moment donné.

L'utilisation d'hypertexte est une solution. Les hypertextes permettent d'atteindre, à partir d'une information, toutes les informations corrélées ; ils fournissent un mode de lecture non linéaire.

Guidage et gestion des erreurs

Il est nécessaire de faciliter l'exploration de l'utilisateur, son apprentissage par l'essai et par l'erreur.

Le guidage est l'ensemble des moyens mis à la disposition de l'utilisateur pour lui permettre de connaître l'état du système et d'établir les liens de causalité entre ses actions et l'état du système.

Le guidage peut être implicite par la présentation et l'organisation des informations affichées de manière à ce que l'utilisateur n'ait pas de questions à se poser quant à l'utilisation de son interface.

Des éléments participant au guidage :

- la présence d'une aide en ligne,
- n'autoriser à un moment donné que les actions possibles, mais permettre à l'utilisateur de comprendre pourquoi seules ces actions sont autorisées,
- permettre de revenir sur les actions,
- proposer des messages d'erreurs clairs et explicites, pour une gestion intelligente des erreurs.

Homogénéité

C'est la capacité d'un système informatique à conserver une logique d'usage constante au sein d'une application, entre applications. Ce critère permet notamment de réduire le temps de recherche de l'information. Il faut toujours essayer de concevoir des applications ayant même apparence et comportement (*look and feel*) :

- à contexte similaire, procédure similaire,
- utilisation de guides de style,
- utilisation de boîtes de dialogue communes...

Descriptivité

La descriptivité regroupe la visibilité et la mise à profit de l'intuition.

Il s'agit de proposer une interface en cohérence avec le monde réel, en utilisant les associations d'idées, en s'appuyant sur la notion de métaphore (reconnaître au lieu de se souvenir).

Esthétique

L'aspect esthétique de l'interface peut s'apprécier à travers les éléments suivants :

- Regroupements des objets dans l'espace,
- Choix des couleurs,
- Choix des signaux visuels,
- Représentations tridimensionnelles...

2.D. La stylique des applications graphiques

2.D.1 Introduction

Le développement des applications informatiques en mode graphique nécessite l'apprentissage de nombreuses nouvelles techniques, et, parmi celles-ci, la connaissance de l'environnement graphique.

Connaître à la fois Windows et un environnement de développement est une condition nécessaire mais non suffisante. En effet, le développeur ne peut plus se cantonner à ses outils de professionnel sans se préoccuper de l'environnement de travail habituel de l'utilisateur.

Un des objectifs principaux de l'évolution vers l'informatique client/serveur est de réaliser la convergence entre l'environnement bureautique, espace de travail quotidien des utilisateurs, et l'environnement applicatif, univers naturel des informaticiens. C'est aux interfaces graphiques (et à Windows en particulier) que revient la charge de la réunification.

Or, cette convergence ne pourra être réalisée qu'en obtenant (de la part des développeurs) un haut niveau d'intégration des applications avec l'environnement qui les supporte. Et ce haut niveau d'intégration, les développeurs ne pourront pas l'atteindre s'ils ignorent tout des programmes manipulés quotidiennement par les utilisateurs.

2.D.2 Une entreprise délicate

Avec les interfaces graphiques tout change, et les variations possibles sont si nombreuses et si subtiles que l'on frôle constamment "l'explosion combinatoire". Le traditionnel écran à 25 lignes et 80 colonnes disparaît, la pagination séquentielle aussi, les limites sont mêmes reculées au-delà du niveau de perception du développeur ou de l'utilisateur.

Les possibilités de mise en oeuvre sont désormais immenses, et les risques d'erreurs ont grandi d'autant.

2.D.3 Une phase nouvelle dans la conception des projets

Autrefois, les erreurs de stylique étaient moins nombreuses et moins pénalisantes. En raison du peu de moyens disponibles, la différence entre une stylique soignée et une ergonomie médiocre était quasiment gommée.

Aujourd'hui c'est l'inverse, une stylique inadéquate et c'est l'échec assuré. Il convient donc de se pencher avec une attention renouvelée sur le problème posé par la relation avec l'utilisateur au travers de l'interface.

La stylique de l'application devient une **pièce angulaire de la phase conceptuelle** de tout développement d'application sous Windows.

Dans cette démarche périlleuse, des règles existent. Il faut les connaître et les respecter.

2.D.4 Un processus itératif

Bien que l'on puisse définir des étapes dans la construction d'une interface pour une application dans un environnement graphique, l'enchaînement des étapes n'est ni rigide, ni irréversible.

En effet, le développeur va effectuer plusieurs allers et retours pour affiner son interface en fonction des remarques des utilisateurs.

De fait, il est toujours plus facile d'améliorer un premier jet que de concevoir tout du premier coup.

2.E. Le concept WIMP

Toutes les interfaces graphiques sur le marché (Windows, Presentation Manager, Macintosh, OSF, Openlook, etc.) ont un certain nombre de points communs, ceux-ci se résument au travers du concept "WIMP" (**W**indow, **I**cone, **M**enu, **P**ointeur).

W comme Window






Les interfaces graphiques permettent un affichage multi-fenêtres qui rend possible le travail sur plusieurs applications à la fois, elles sont l'expression visible du multitâches.

Les fenêtres applicatives sont toujours non modales (Modal = mode de fonctionnement, un dialogue modal enferme l'utilisateur dans un mode de fonctionnement).

I comme Icône

Un schéma vaut mieux qu'un long discours ! Les objets sont représentés par des icônes (petits dessins) plutôt que par des libellés.

Quelques exemples à partir de la barre d'outils de Word 7.

Icône	Signification
	Nouveau document
	Pour ouvrir un document
	Disquette : pour enregistrer un document
	Imprimante : pour imprimer
	Loupe : pour aperçu avant impression

M comme Menu

L'emploi de menu permet de proposer un processus basé sur l'association objet/action :

Le processus objet-action est le mode d'interaction avec l'interface, il se décompose en trois phases :

1. désignation de ou des objets (ex : choix d'un paragraphe dans un texte),
2. choix de l'action (ex : l'option couper - mettre dans le presse papier - du menu édition) et enfin,
3. production du résultat (Résultat : le paragraphe a disparu dans le texte, sans doute pour être placée ailleurs).

P comme Pointeur

Le pointeur (le plus souvent une souris) est l'élément permettant de désigner les objets, choisir les menus ou changer de contexte en passant d'une fenêtre à l'autre.

2.F. La norme CUA

CUA (Common User Access ou interface commune d'accès) est une norme proposée par IBM pour le design d'application en environnement graphique.

Le respect de cette norme doit permettre d'obtenir une bonne facilité d'utilisation et surtout une cohérence (une interface est cohérente si elle respecte à tout instant des standards de présentation et de comportement) avec l'environnement graphique utilisé à l'intérieur de toute application.

Cette cohérence permettra également une bonne intégration de l'application dans l'environnement vis-à-vis des autres applications (votre application s'utilisera comme les autres applications disponibles sur le marché pour les environnements graphiques Windows).

Les deux principes essentiels de la stylisme de CUA sont .

Principe 1 : Reproduire le modèle conceptuel de l'utilisateur.

- Employer des métaphores (icône de l'imprimante)
- Faire ressembler l'interface au domaine de l'utilisateur

Exemple : calculatrice sous Windows.

Principe 2 : L'utilisateur a le contrôle

C'est l'utilisateur qui contrôle l'application et non l'application qui contrôle l'utilisateur.

Les techniques qui permettent de rendre le contrôle du dialogue à l'utilisateur sont les suivantes :

- Faire une interface indulgente : pardonner les erreurs éventuelles de l'utilisateur. Pour cela les actions doivent être facilement annulables et celles qui ne le sont pas doivent obtenir confirmation avant continuation.
- Faire une interface visuelle : les utilisateurs doivent plutôt voir que se rappeler comment procéder pour utiliser l'application.
- Donner un retour d'information immédiat.

2.G. La conception des interfaces

Le développement d'une interface homme-machine doit être basé sur la vision de l'utilisateur.

Les enjeux du processus

La conception des interfaces devient aujourd'hui un véritable projet qui représente jusqu'à près de 50 % du logiciel réalisé selon une étude américaine.

Il est nécessaire que l'application soit adaptée aux utilisateurs (interface « sur mesure »).

Outre le non-respect des critères ergonomiques, un certain nombre d'erreurs doivent être évitées :

- le fait de penser que si l'interface satisfait l'informaticien, elle doit obligatoirement satisfaire l'utilisateur final.
- lorsqu'une application ne satisfait l'utilisateur, celle-ci risque d'être revue et de perdre de la cohérence au fil des modifications.
- la conception des interfaces ne doit être guidée par des critères de performances.
- le regroupement par fonctionnalités plutôt que selon la méthode de travail de l'utilisateur risque de ne pas être satisfaisant.

Il est donc essentiel de bien connaître l'utilisateur et les tâches qu'il doit réaliser à l'aide de l'application.

Les acteurs du processus sont :

- **l'utilisateur** : l'interface doit prendre en compte son expérience par rapport à l'outil.
- **l'ergonome** : qui sert d'interface entre l'utilisateur et le concepteur dans la définition du dialogue homme-machine. Il étudie l'activité de l'utilisateur dans un univers de référence.
- **le concepteur** : qui réalise l'interface.

D'autres acteurs peuvent intervenir dans le processus par exemple les cognitivistes, les infographistes...

Méthodologie de conception

La conception s'effectue en plusieurs étapes.

Analyse de la demande

Le domaine des interfaces est pluridisciplinaire. Il s'agit dans un premier temps de déterminer le degré d'intervention de chaque acteur dans le projet et de définir le protocole de collaboration.

Analyse de l'existant

C'est typiquement le travail de l'ergonome. Il s'agit de déterminer comment travaille l'utilisateur, quelles sont les procédures et les consignes qu'il suit dans l'accomplissement de chaque tâche et dans quel environnement. Cette analyse doit recenser les informations et les fonctions constituant la tâche, le séquençement opérationnel entre événements, données, décisions, actions...

Définition des besoins

Les auteurs insistent sur la nécessité de nettement dissocier la phase d'analyse de l'existant de la phase de définition des besoins.

Le besoin doit être exprimé de manière indépendante de toute solution.

L'expression des besoins est structurée suivant les tâches nécessaires pour l'utilisateur en reflétant l'organisation de son travail (graphe des tâches) et précise pour chaque tâche les exigences et les performances.

Il doit y avoir séparation des exigences portant sur l'interface homme-machine de celle portant sur l'applicatif (conception d'applications séparant les deux niveaux).

Au niveau du dialogue définissant la relation entre l'utilisateur et la machine, il faut déterminer le type de dialogue à mettre en place. Faut-il laisser l'initiative du dialogue à l'utilisateur ? à la machine ? ou aux deux ?

lorsque le dialogue est à l'initiative de l'utilisateur, celui-ci détermine son propre scénario pour réaliser une tâche. Ce type de dialogue est possible pour un utilisateur expérimenté. Les dialogues à l'initiative de la machine sont préférés pour les utilisateurs novices ou occasionnels.

Maquettage et prototypage

Reproduction à l'échelle réduite de l'application, la maquette contient la plupart des éléments de base : ensemble des fenêtres, des dialogues et des différents messages.

2.H. Le maquetage

La maquette de l'application est l'une des étapes nouvelles du développement d'applications sous Windows. Auparavant, les efforts étaient concentrés sur les données et les traitements (cf. MERISE).

2.H.1 *Maquette et prototype*

La maquette représente plutôt une fonction horizontale, tandis que le prototype une fonction verticale.

La maquette contient l'ensemble des fenêtres et dialogues de l'application, mais rien n'est codé derrière ceux-ci (si ce n'est les enchaînements). Il n'y a pas d'accès aux données, pas de calcul.

Le prototype, quant à lui, est le développement complet d'une partie de l'application, effectué dans le but d'une mesure. Il contient non seulement le ou les fenêtres et/ou dialogues, mais bien sûr le code permettant d'accéder aux données. le prototype est par exemple développé pour valider une solution technique pour vérifier la faisabilité ou bien pour mesurer un temps de réponse.

2.H.2 *Le contenu de la maquette*

La maquette de l'application doit être complète.

Elle contient toutes les fenêtres, le ou les menus, les dialogues, les messages d'erreurs et bien sûr la cinématique de l'application, c'est-à-dire les enchaînements des fenêtres et des dialogues.

Entre la maquette et le produit terminé, il n'existe qu'un écart très faible (10% en général. Le rôle de la maquette est prépondérant dans le projet.

La maquette permet de **concevoir l'ergonomie générale de l'application**, de montrer à l'utilisateur comment on a imaginé de mettre en oeuvre une fonction particulière, comment on a résolu tel ou tel problème, et de présenter enfin à l'utilisateur ce que sera la future application.

La maquette permet aussi de figer l'apparence et le comportement ("*look and feel*") de l'application. Elle permet également de responsabiliser l'utilisateur, en l'impliquant naturellement au processus de développement.

La maquette est "le modèle", le "sur mesure" que le concepteur va fabriquer pour l'utilisateur. Le client (l'utilisateur) doit "essayer" la maquette, pour vérifier l'adéquation entre les fonctions qu'il a demandées et le modèle qui lui est présenté.

L'utilisateur doit obligatoirement **valider la maquette**, car la maquette est le point de départ de la phase de conception détaillée, et de réalisation.

Dans le cycle de vie d'un projet, la phase de maquettage s'inscrit entre la phase de conception générale et la conception détaillée.

Grâce aux fonctions de l'application qui ont été énumérées et qualifiées dans la phase de conception générale, le concepteur va pouvoir construire le menu qui est le point d'entrée de l'application.

La réalisation de la maquette peut s'effectuer selon les phases suivantes :

1. Définition des normes : normes de notation, taille des boutons OK et Annuler, touches de fonction, raccourcis clavier; etc.
2. Maquettage dans l'ordre des éléments suivants
 1. La fenêtre principale,
 2. Les menus contextuels,
 3. Les fenêtres filles,
 4. Les boîtes de dialogues.
3. Pour ensuite effectuer
 1. La "recette" de la maquette
 2. La présentation de la maquette aux utilisateurs

Il faut **procéder par itération** et organiser des présentations à l'utilisateur dès que le coeur de l'application est maquetée.

L'utilisateur doit concentrer son activité de validation sur les points suivants :

1. le menu, la navigation et l'enchaînement des dialogues,
2. le contenu des dialogues et des fenêtres,
3. l'accès évident aux fonctionnalités de l'application.

La maquette permet de valider la conception sur les plans ergonomique et fonctionnel, le prototype permet de vérifier la faisabilité technique des exigences demandées.

Sur de petits projets, le développement de l'interface peut se faire avec l'utilisateur selon une démarche itérative et incrémentale (RAD).

3. LES NORMES DE PRÉSENTATION

3.A. Des applications Windows

Pour réussir le développement de vos applications informatiques, la connaissance des principes de fonctionnement d'une interface graphique et des règles de style est incontournable. Il est nécessaire de comprendre ces principes, et de savoir les appliquer à travers l'ensemble des objets (les contrôles) offerts par une interface graphique : fenêtres, boîtes de dialogues, widgets...

La conception de l'interface utilisateur devient donc un des points-clés de la réussite d'un projet informatique.

3.A.1 *Les composants d'une interface graphique*

Avant de présenter les principales règles qui régissent la stylistique des applications sous Windows 3.1 et Windows 95, il est nécessaire de pouvoir situer les différents composants de l'interface graphique et de les identifier par leurs appellations exactes.

Les différents objets graphiques sont regroupés en 3 catégories :

1. Les contrôles (ou "Widget")
2. Les fenêtres
3. Les menus

3.A.2 *Les contrôles (ou "Widget")*


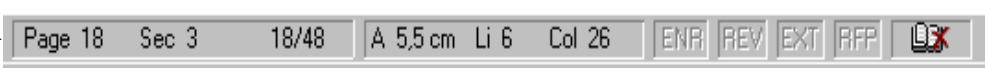

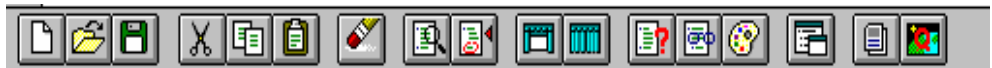

a. Contenu

Les contrôles sont tous les objets graphiques qui composent les fenêtres et dialogues, de la boîte à cocher jusqu'à l'ascenseur vertical. Leur nombre et leur type ne sont pas limités.


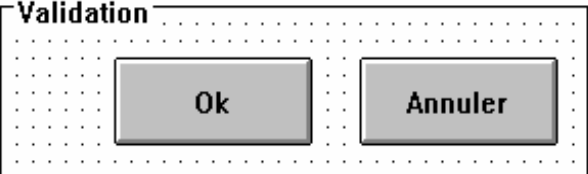
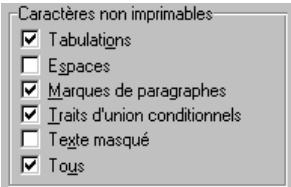

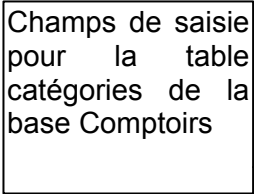
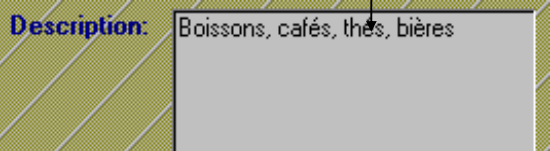
Les contrôles non standard sont aussi appelés "Custom Controls".

Le terme "contrôle" désigne les composants de l'interface graphique. C'est un terme générique qui identifie aussi bien les fenêtres, les menus que les objets élémentaires de l'interface : liste déroulante, case à cocher...

SOMMAIRE

Nom du contrôle	Caractéristiques
Barre de défilement, ascenseur 	<p>Permettre le déplacement dans un objet ou une fenêtre.</p> <p>Lorsqu'on clique sur la flèche ou dans le puits, le "contrepoids" se déplace, et le contenu de l'objet associé suit le déplacement.</p>
Barre d'état <div>Barre de Word 7</div> 	<p>Donner des informations contextuelles sur l'état de l'application.</p> <p>Affiche, par exemple, les informations concernant un document Word, dans une barre constituée de champs d'édition fixe.</p>
Barre de message ou micro-help 	<p>Donner des informations contextuelles sur les fonctions de l'application, des menus...</p> <p>Affiche une ligne de texte, par exemple en fonction de l'item de menu sélectionné.</p> <p>Exemple : ligne d'aide du menu Fichier de Word</p> <div>Crée, ouvre, enregistre, imprime des documents, ou quitte Word</div>
Barre d'outils, palettes, règles, ruban	<p>Fournir des raccourcis visuels aux actions réalisables par menus.</p> <p>Les comportements de la barre d'outils et du ruban sont fonction des objets qui les composent.</p> <div style="text-align: center;"> <p>Barre d'outils de SqlWindows</p>  <p>Ruban de Word 7</p>  </div>
Bouton de commande ou bouton d'action	<p>Permettre de déclencher une action : navigation, changement d'état...</p> <p>Lorsqu'on clique sur le bouton, il s'enfonce (l'aspect graphique change), puis l'action associée au bouton est exécutée, quand on relâche.</p> <div> <div> <p>Boutons de type texte :</p> <div>OKAnnuler</div> </div> <div> <p>Boutons de type image :</p> <div>?</div> </div> <div> <p>Boutons de type contrôle :</p> <div>↑↓</div> </div> </div>

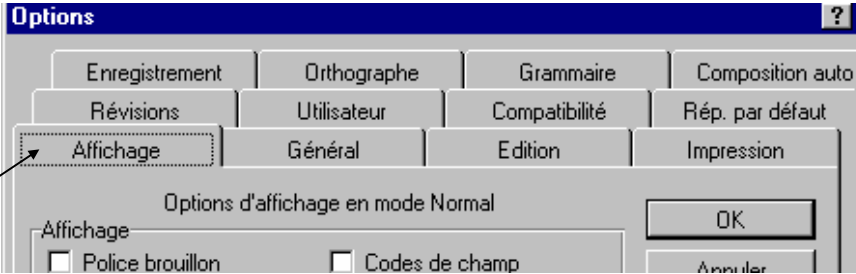
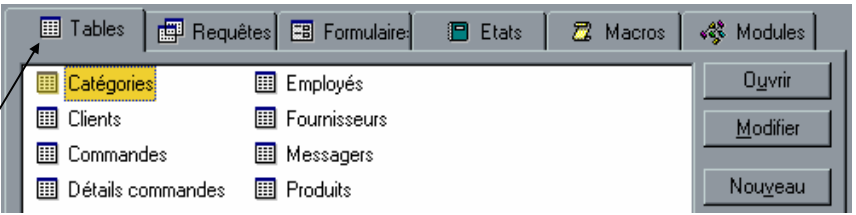
SOMMAIRE

Bouton défilant, molette de réglage 	<p>Modifier une valeur sans saisie.</p> <p>Cliquer sur une des deux flèches modifie la valeur du champ d'édition associé.</p>
Zone de groupe, boîte de regroupement	<p>Regrouper des contrôles logiquement dépendants (ex : un ensemble de boutons poussoirs permettant de choisir une commande).</p> <p>Délimiter les frontières d'un groupe d'options.</p> 
Case à cocher, boîte à cocher 	<p>Utilisée pour saisir ou visualiser des choix multiples, non exclusifs les uns des autres. Affiche clairement si l'option décrite dans le libellé associé à cette boîte à cocher est active (présence d'une croix dans la boîte) ou inactive (pas de croix) ou encore parfois, dans un état incertain.</p> <p>Le fait de cliquer sur la boîte ou sur le libellé de la boîte rend l'option active ou inactive suivant le principe de la bascule.</p>
Case d'option, bouton radio 	<p>Utilisée pour saisir ou visualiser des choix multiples, exclusifs les uns des autres. Affiche clairement si l'option décrite dans le libellé associé à ce bouton-radio est active (présence d'un point dans le cercle) ou inactive (pas de point), un bouton-radio est toujours associé à, au minimum, un autre bouton-radio de façon exclusive.</p> <p>Le bouton-radio est utilisé pour des choix exclusifs. Le fait de cliquer sur le bouton ou sur le libellé du bouton rend l'option active et désactive du même coup le bouton associé qui détenait la valeur "vrai" auparavant (principe de la bascule).</p>
Libellé, champ d'affichage	<p>Informar sur le contenu d'un champ d'édition ou champ d'édition en affichage.</p> <p>Toujours justifié à gauche.</p>
Zone de texte, champ de saisie 	<p>Permettre la saisie de texte. La zone de texte peut être multi-lignes et comporter un ascenseur.</p> <p>Le curseur prend la forme d'une barre verticale (I) lorsqu'on peut saisir. Il dispose d'un équivalent clavier (souligné) permettant un accès direct au champ d'édition par combinaison de la touche Alt + l'équivalent clavier.</p> 

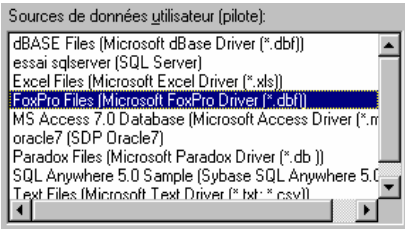
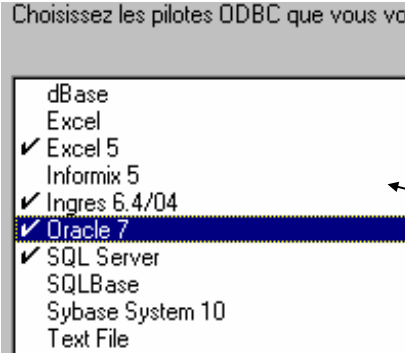
SOMMAIRE

<p>Zone de liste, liste de choix (List Box)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> Affichage de la liste des noms villes de localisation des avions. </div>	<p>Permettre la sélection simple ou multiple dans une liste d'éléments. La zone de liste est utilisée pour la saisie guidée.</p> <p>Lorsqu'on sélectionne sur un élément de la liste, celui-ci apparaît en vidéo inverse. Quand on clique sur les flèches de l'ascenseur, le contenu de la liste défile.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>Liste des localisations :</p> <div style="border: 1px solid black; height: 50px; width: 100%;"></div> </div>
<p>Zone de liste déroulante (Drop Down List)</p>	<p>Permettre la sélection simple dans une liste d'éléments.</p> <p>Similaire à la zone de liste, cependant le contenu de la liste n'est pas affiché en permanence, seulement lorsqu'on clique sur la flèche.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> Imprimer: Document <div style="border: 1px solid black; padding: 2px;"> Document Résumé Annotations Styles Insertions automatiques Affectation des touches </div> </div>
<p>Zone de liste modifiable (Combo Box)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> Capacité : <div style="border: 1px solid black; width: 50px; height: 15px; display: inline-block;"></div> </div>	<p>Permettre la sélection simple dans une liste d'éléments, ou la saisie directe.</p> <p>Comme une zone de liste pour la partie inférieure, comme un champ d'édition pour la partie supérieure.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>Combo-Box: cmb1</p> <p>→ List-Initialization</p> <p>→ → Text: 150</p> <p>→ → Text: 170</p> <p>→ → Text: 190</p> <p>→ → Text: 210</p> <p>→ → Text: 230</p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px; width: fit-content;"> Initialisation de la liste des capacités avec 5 valeurs. Syntaxe SqlWindows </div>
<p>Zone de liste déroulante modifiable (Drop Down Combo Box)</p>	<p>Permettre la sélection simple dans une liste d'éléments, ou la saisie directe.</p> <p>Similaire à la zone de liste modifiable, mais le contenu de la liste n'est pas affiché en permanence, seulement lorsqu'on clique sur la flèche. Visuellement, la combo se différencie de la drop down simple par la présence d'un écart entre le champ de saisie et la flèche d'affichage de la liste.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>State: CO</p> <div style="border: 1px solid black; padding: 2px;"> OH Ohio OK Oklahoma OR Oregon </div> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px; width: fit-content;"> Liste de sélection des noms des états des USA </div>

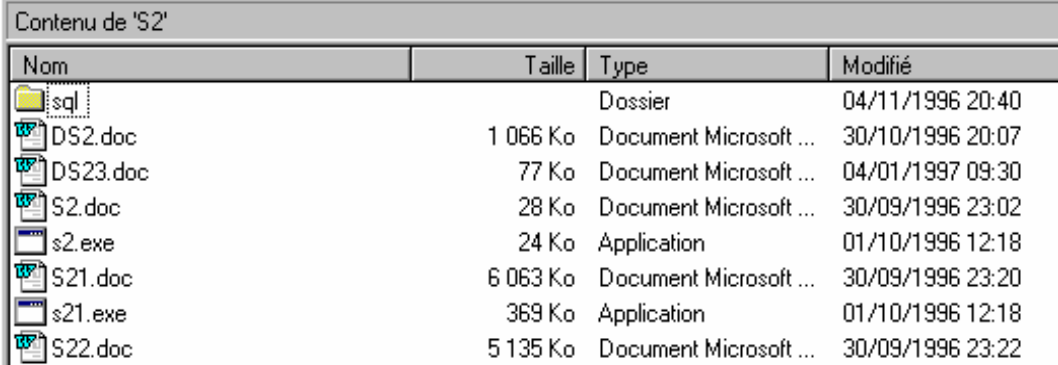
SOMMAIRE

<p>Onglet</p> <p>Permettre de définir des affichages logiques répartis sur plusieurs pages à l'intérieur d'une même fenêtre.</p> <p>Un clic sur un onglet affiche les informations associées à cet onglet.</p>	
<p>Word6 pour Windows 3.1 et 95</p>	
<p>Onglet Affichage activé</p>	
<p>Access 7 pour Windows 95</p>	
<p>Onglet Tables activé</p>	

Windows 95 se caractérise d'une part, par un changement important au niveau de l'apparence des composants (se reporter à L'interface de Windows 95 présentée dans la séquence 1 du cours sur l'architecture logicielle des systèmes informatiques) et d'autre part, par des nouveaux contrôles graphiques présentés ci-dessous.

<p>Liste de choix mono-sélection</p> 	<p>Permettre la sélection simple dans une liste d'éléments.</p> <p>Lorsqu'on sélectionne un élément de la liste, celui-ci apparaît en vidéo inverse. Quand on clique sur les flèches d'ascenseur, le contenu défile.</p>
<p>Liste de choix à multiple sélection</p> <p>Choisissez les pilotes ODBC que vous voulez installer :</p> 	<p>Permettre la sélection multiple dans une liste d'éléments.</p> <p>Lorsque l'on sélectionne un élément de la liste, une marque apparaît dans la boîte associée à l'élément de la liste. Le clic sur la boîte ou sur le libellé d'un élément de la liste rend la sélection active ou inactive suivant le principe de la bascule.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Liste de choix des pilotes ODBC à installer pour AMC*Designer.</p> <p>Les 4 pilotes sélectionnés sont repérés par la marque : </p> </div>

SOMMAIRE

Liste de choix visuel	<p>Permettre la sélection simple dans une liste d'éléments caractérisés par une image et un texte. Une liste de choix dispose de 3 modes d'affichage : une vue par icône standard (32x32), une vue par icône réduite et texte, une vue par liste (détaillée ou non).</p> <p>Lorsqu'on sélectionne un élément de la liste, le libellé associé à l'image apparaît en vidéo inverse. Quand on clique sur les flèches d'ascenseur, le contenu défile.</p>
	
Liste de choix hiérarchique	<p>Permettre la visualisation et la navigation à l'intérieur d'une arborescence.</p> <p>A l'initialisation, la liste contient tous les éléments de premier niveau.</p> <p>Le clic sur + ouvre l'arborescence de l'élément sélectionné.</p> <p>Un clic sur - referme l'arborescence de l'élément sélectionné.</p>

b. Règles sur les contrôles

Règle C1	La largeur des boutons d'actions est unique pour l'ensemble des boutons d'un dialogue, ou à défaut, pour les boutons d'un même groupe.
Règle C2	Les libellés des boutons d'actions doivent être accompagnés d'un raccourci clavier permettant leur activation par la combinaison de touches Alt + la lettre choisie.
Règle C3	Le libellé "OK" est toujours intégralement en majuscule (les deux lettres), sans souligné.
Règle C4	Le libellé Annuler est rédigé comme un nom propre (1ère lettre en majuscule), sans souligné.
Règle C5	Un bouton d'action qui provoque l'ouverture d'un dialogue est toujours libellé avec des points de suite (...).
Règle C6	Dans les boutons mixtes (texte+icône), l'écart entre les deux éléments est de 4 pixels dans tous les cas (horizontal et vertical).
Règle C7	Un libellé qualifiant un contrôle dont le contenu est modifiable comprend obligatoirement un raccourci clavier.
Règle C8	Un libellé qualifiant un contrôle dont le contenu n'est pas modifiable ne comprend pas de raccourci clavier.
Règle C9	Le libellé qualifiant une zone de texte est obligatoirement suivi du signe deux points (:), sans espace avant et après.
Règle C10	Les libellés qualifiant un contrôle peuvent comprendre plusieurs mots, mais seul le premier mot sera rédigé comme un nom propre (1ère lettre en majuscule et seulement celle-là).
Règle C11	Combinaisons des touches standards dans les champs de saisie: - Ctrl + Home : retour au début du champ, - Ctrl + End : accès à la fin du champ.
Règle C12	Combinaisons de touches standards dans les champs de saisie multi-lignes : - Home : retour au début de la ligne courante, - End : accès à la fin de la ligne courante.
Règle C13	Combinaisons de touches standards dans une zone de liste : - PgUp : accès à la première ligne visible de la liste, - PgDn : accès à la dernière ligne visible de la liste, - Ctrl + PgUp : accès à la première ligne de la liste, - Ctrl + PgDn : accès à la dernière ligne de la liste.

3.A.3 Les Fenêtres

a. Contenu

La fenêtre est la principale interface entre l'utilisateur et une application dotée d'une interface graphique. Elle permet à l'utilisateur d'agir sur l'application.

Cette catégorie comprend les fenêtres et les objets qui les composent (barre de titre, case système...) ainsi que les boîtes de dialogues.

Une boîte de dialogue est une fenêtre particulière, utilisée pour "dialoguer" ou pour communiquer avec l'utilisateur (dialogue de saisie, dialogue de choix, dialogue d'information, dialogue de message,...).

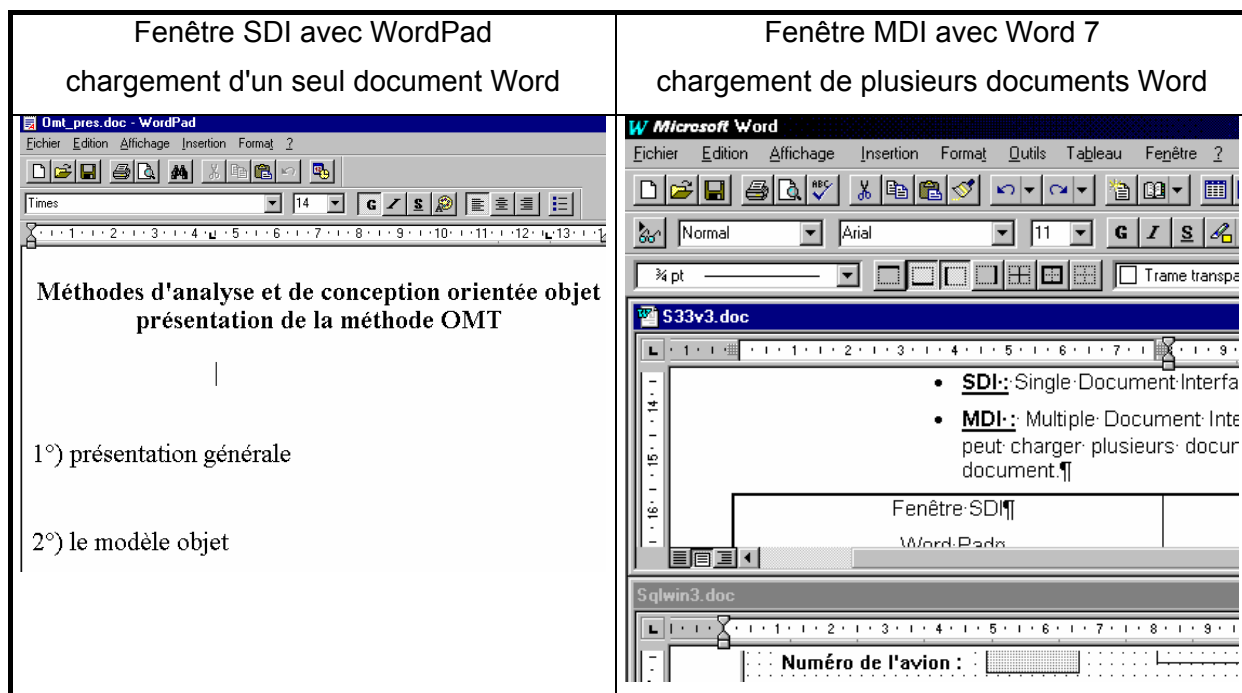
La boîte de dialogue diffère de la fenêtre car elle est le plus souvent **modale** (l'utilisateur doit répondre au dialogue avant de continuer l'application). Elle n'est pas redimensionnable, et ne contient pas les mêmes contrôles que les fenêtres (une boîte de dialogue ne contient pas de barre d'outils par exemple).

Fenêtre d'application, fenêtre principale

Cette fenêtre délimite l'espace de travail de l'application. Toutes les fenêtres de l'application vont venir s'ouvrir à l'intérieur de cette fenêtre sauf les menus si la fenêtre a été retaillée, les boîtes de dialogues et l'aide en ligne de l'application)

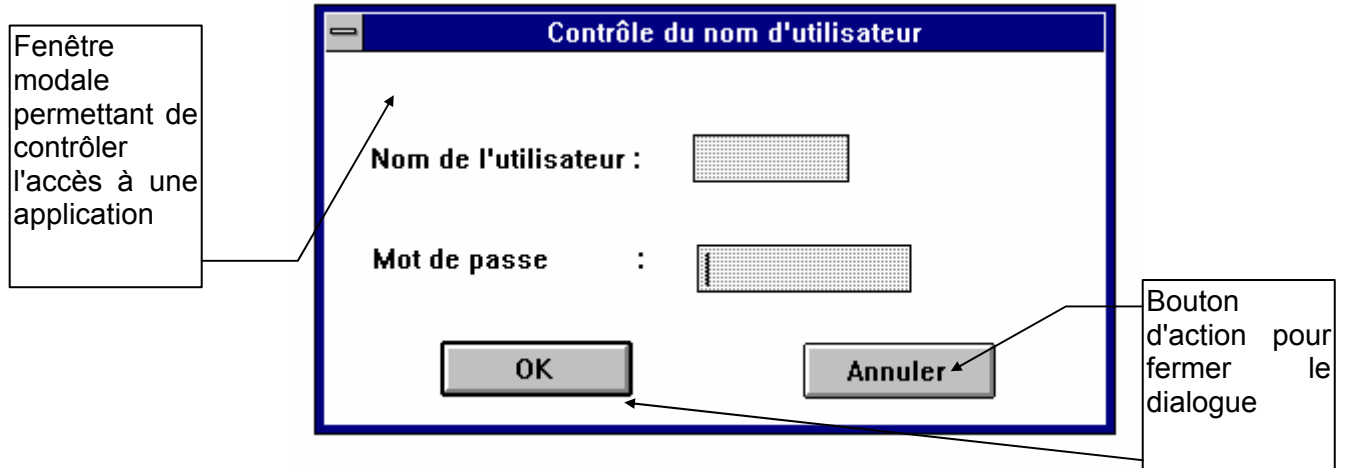
La fenêtre principale peut être déplacée et retaillée. Elle est toujours non modale. Elle peut être du type :

- **SDI** : Single Document Interface (WordPad)
- **MDI** : Multiple Document Interface (Word 7) L'application peut charger plusieurs documents ou plusieurs instances d'un même document.



Boîte de dialogue

Son rôle est de permettre à l'utilisateur la saisie d'informations ou la demande d'informations complémentaires. Une boîte de dialogue peut être déplacée, elle est toujours modale (sauf pour les boîtes de dialogues de recherche), et elle contient au minimum un bouton d'action pour fermer le dialogue.



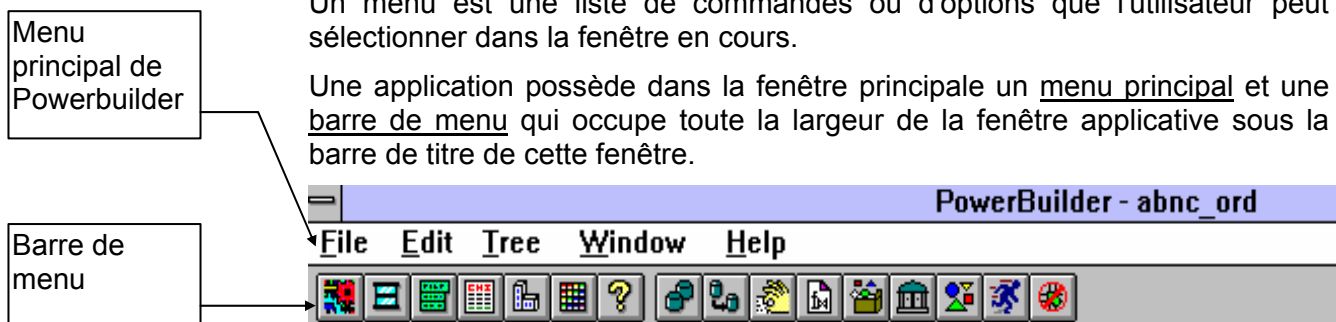
b. Règles sur les fenêtres

Règle F1	Le libellé dans la barre de titre est toujours centré pour Windows 3.1 et cadré à gauche pour .Windows 95.
Règle F2	Les fenêtres applicatives ou fenêtres filles MDI sont toujours non modales.
Règle F3	Les fenêtres filles MDI sont dépourvues de boutons, de barre de menus, de barre d'outils et de barre d'état.
Règle F4	La fenêtre principale d'une application est retailable par les bords, comporte un menu système, une barre de titre, des boutons Min et Max (mise en icône et plein écran) et est non modale.
Règle F5	La barre d'outils doit prendre toute la largeur de la fenêtre applicative quelle que soit la dimension de cette dernière.
Règle F6	La barre d'état doit prendre toute la largeur de la fenêtre applicative quelle que soit la dimension de cette dernière.
Règle F7	La fenêtre principale contient un et un seul menu.

3.A.4 Les Menus

Un menu est une liste de commandes ou d'options que l'utilisateur peut sélectionner dans la fenêtre en cours.

Une application possède dans la fenêtre principale un menu principal et une barre de menu qui occupe toute la largeur de la fenêtre applicative sous la barre de titre de cette fenêtre.



Cette catégorie est constituée des différents types de menus (menus déroulants, menu en cascade...), ainsi que les objets propres des menus (items, coche...).

a. Les trois types de menus d'une application

1. Les menus déroulants (Drop Down), sont les plus utilisés. Ils apparaissent dans la fenêtre principale de l'application

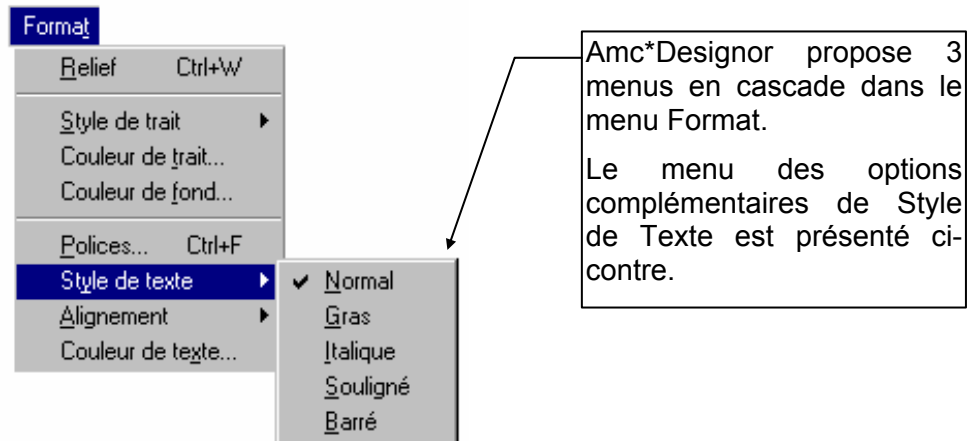
Un menu déroulant offre un ensemble d'items (équivalent à lignes, options ou éléments) déclenchant soit une action, soit un dialogue. Il s'ouvre quand on clique sur le libellé de la barre de menu.

Menus déroulants de la fenêtre principale de Visual Basic 4	Menus déroulants de la fenêtre principale de l'application Location de matériels Génération avec Amc*Designor pour Visual Basic
Options du menu Fichier	Options du menu Rubriques

SOMMAIRE

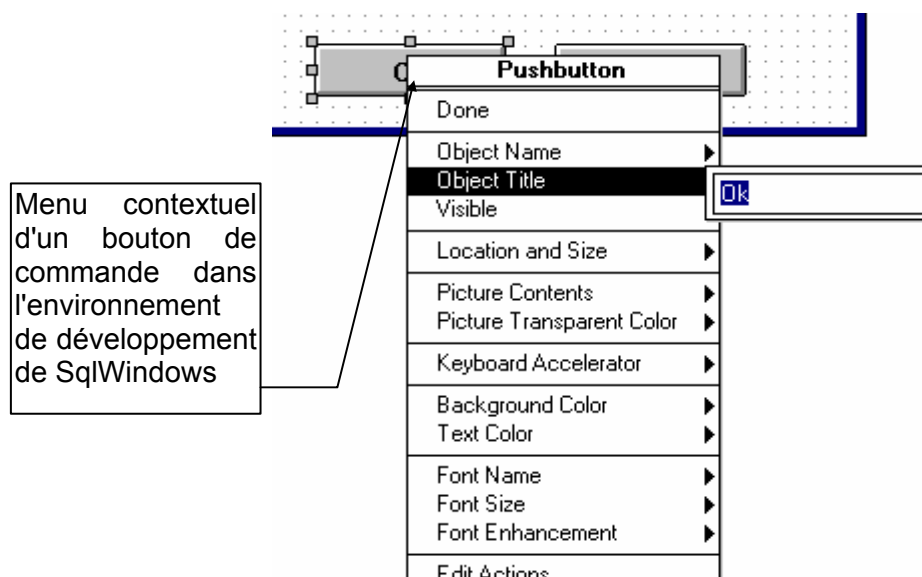
2. Les **menus en cascade**, sont des sous-menus attachés à la droite des items des menus. Ils sont repérés par un triangle ▶.

Un menu en cascade offre un ensemble d'options complémentaires à un item de menu classique. Il s'ouvre quand on sélectionne un item de menu suivi du symbole triangle.



3. Les **menus popup**, dénommés également menus flottants ou menus contextuels, sont placés au plus près d'un objet, d'un bouton d'action ou d'une table par exemple. Les menus popup présentent des options spécifiques à l'objet.

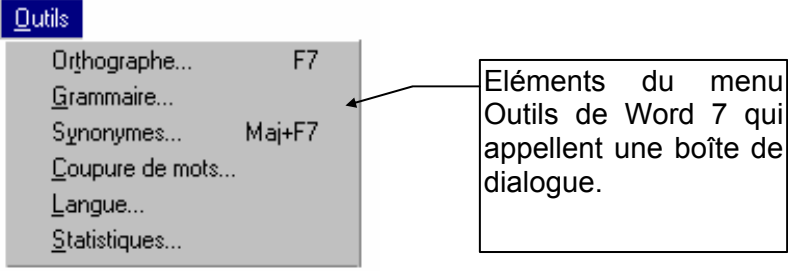
Ils permettent de proposer des items déclenchant des actions ou des dialogues, hors d'une barre de menu. Ces menus contextuels sont généralement accessibles par le bouton droit de la souris, et s'affichent à l'endroit où est positionné le curseur.



b. Règles sur les menus

Règle M1	Le menu Fichier est obligatoire dans toutes les applications, il se place à gauche sur la barre de menu (premier menu) et contient au minimum l'item Quitter avec son équivalent clavier (Alt+F4).																		
Règle M2	La barre de menu d'une application MDI contient au minimum les items Fichier, Edition, Fenêtre et ? (aide).																		
Règle M3	Le menu Edition est exclusivement réservé à la gestion du presse-papier.																		
Règle M4	<p>Le menu Edition, si nécessaire, doit contenir les items suivants et respecter les raccourcis clavier standards.</p> <p>Annuler Ctrl + Z, Couper : Ctrl + X , Copier : Ctrl + C</p> <p>Coller : Ctrl + V, Effacer : Suppr</p> <div><div>Edition</div><table><tbody><tr><td>Annuler</td><td>Ctrl+Z</td></tr><tr><td>Répéter</td><td>F4</td></tr><tr><td>Couper</td><td>Ctrl+X</td></tr><tr><td>Copier</td><td>Ctrl+C</td></tr><tr><td>Coller</td><td>Ctrl+V</td></tr><tr><td>Effacer la sélection</td><td>Suppr</td></tr><tr><td>Sélectionner tout</td><td>Ctrl+L</td></tr><tr><td colspan="2">Copier vers...</td></tr><tr><td colspan="2">Coller à partir de...</td></tr></tbody></table><div>Menu Edition de Paint</div></div>	Annuler	Ctrl+Z	Répéter	F4	Couper	Ctrl+X	Copier	Ctrl+C	Coller	Ctrl+V	Effacer la sélection	Suppr	Sélectionner tout	Ctrl+L	Copier vers...		Coller à partir de...	
Annuler	Ctrl+Z																		
Répéter	F4																		
Couper	Ctrl+X																		
Copier	Ctrl+C																		
Coller	Ctrl+V																		
Effacer la sélection	Suppr																		
Sélectionner tout	Ctrl+L																		
Copier vers...																			
Coller à partir de...																			
Règle M5	La fenêtre principale contient un et un seul menu. Il peut être modifié en fonction du contexte applicatif.																		
Règle M6	Tous les menus ainsi que les items des menus sont dotés d'un équivalent clavier rappelé par une lettre soulignée.																		
Règle M7	Les boutons d'actions d'une barre d'outils ont obligatoirement un équivalent dans les menus.																		
Règle M8	<p>Si les items du menu Fichier sont obligatoires, il faut leur associer les équivalents clavier suivant :</p> <table><tbody><tr><td>Nouveau</td><td>Ctrl + N</td><td>Ouvrir</td><td>Ctrl + O,</td></tr><tr><td>Enregistrer</td><td>Ctrl + S</td><td>Imprimer</td><td>Ctrl + P</td></tr></tbody></table>	Nouveau	Ctrl + N	Ouvrir	Ctrl + O,	Enregistrer	Ctrl + S	Imprimer	Ctrl + P										
Nouveau	Ctrl + N	Ouvrir	Ctrl + O,																
Enregistrer	Ctrl + S	Imprimer	Ctrl + P																
Règle M9	Le menu Aide (?) contient au minimum un item d'accès à l'index de l'aide (Index) et un item d'accès au dialogue affichant la signature du logiciel (A propos de...).																		
Règle M10	Si l'application gère une aide contextuelle, le menu d'aide (?) contient un item d'accès à l'aide sur la fenêtre active (Aide sur...).																		
Règle M11	Dans une application MDI, il y a toujours un menu Fenêtre avec les items suivants : Cascade, Mosaïque , Réorganiser les icones, un séparateur puis la liste des fenêtres filles ouvertes.																		

SOMMAIRE

Règle M12	Dans une application MDI, s'il y a plus de 10 fenêtres ouvertes, le menu Fenêtre contient la liste des 9 dernières fenêtres filles utilisées et un item permettant d'accéder à la liste des fenêtres filles ouvertes (Suite des fenêtres...).
Règle M13	<p>Les items de menus qui provoquent l'affichage d'une boîte de dialogue sont suivis des points de suite (...)</p>  <p>Eléments du menu Outils de Word 7 qui appellent une boîte de dialogue.</p>
Règle M14	Il ne faut pas isoler un item de menu entre deux barres de séparation sauf si celui-ci est placé en première ou en dernière position.
Règle M15	Le libellé d'un menu est toujours en un seul mot dans la barre de menus.
Règle M16	Le libellé d'un menu est toujours rédigé comme un nom propre (1ère lettre en majuscule et seulement celle-là).
Règle M17	Les libellés des items dans les menus ne sont pas limités à un seul mot mais ne doivent pas dépasser quatre mots (article compris), et seul le premier mot du libellé sera rédigé comme un nom propre. Ce libellé comportera au maximum 25 caractères (environ).
Règle M18	Le menu Fenêtre est toujours l'avant dernier à droite sur la barre de menu, juste avant le menu d'aide.
Règle M19	Le menu d'aide (?) est toujours placé en dernière position dans la barre de menu.
Règle M20	Les items d'un menu popup ont obligatoirement leur équivalent dans les différents menus de l'application.

Vérification
des règles
M15, M16,
M18 et M19
dans Word 7



3.B. Des applications Internet et intranet

Le développement d'applications, pour Internet et/ou un intranet, en mode hypertexte est différent du développement d'application en mode graphique.

Pour réussir ces nouveaux types de projets, la connaissance des principes de fonctionnement d'une interface hypertexte et des règles de style est indispensable. Il est nécessaire de comprendre ces principes, et de savoir les appliquer à travers l'ensemble des objets offerts par l'interface hypertexte : documents, liens, images...

3.B.1 De Internet à l'intranet

Internet est le plus important réseau informatique mondial.

L'intranet (concept inventé par Jim Clark de la société Netscape) est un réseau informatique privé basé sur l'utilisation des outils et standards utilisés de l'Internet public pour la mise en place de systèmes d'information d'entreprise.

Une solution intranet se caractérise par la mise en place d'un (ou de plusieurs) serveur(s) intranet (serveur web à usage privé) au sein du réseau local et la mise à disposition d'outils (lecteur de web, messagerie, gestionnaire de "news groups" ...) issus du monde Internet sur les postes des utilisateurs.

3.B.2 Les documents hypertextes ou pages HTML

Un document hypertexte, généralement appelée page HTML, est un document consultable sur le WEB (expression abrégée de "World Wide Web"), sorte d'espace dans lequel les documents permettent de conduire à d'autres documents sans qu'il ne soit nécessaire de tenir compte du lieu où ils résident.

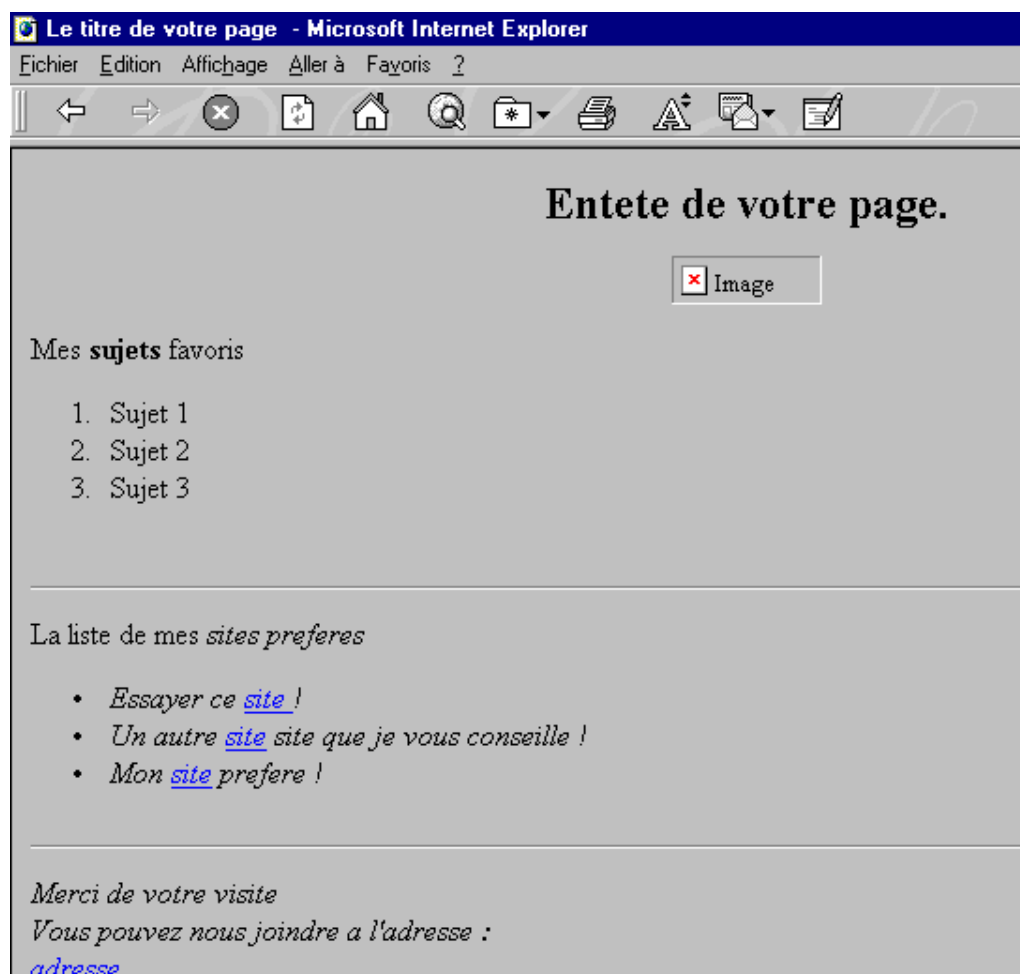
Une page HTML est une page écrite en "HyperText Markup Language", langage consistant en un ensemble de codes permettant à un logiciel de consultation (Navigateur ou "Browser") d'afficher l'information.

HTML est un ensemble simple de commandes de formatage de documents.

Le Web est un système de publication de documents fondé non seulement sur le langage HTML mais également sur un protocole de transfert de documents : le protocole HTTP (HyperText Transfer Protocol).

SOMMAIRE

Exemple d'un gabarit ("template") pour une page d'accueil personnelle



Source HTML de la page d'accueil ci-dessus

```
<HTML>
<HEAD>
<TITLE> Le titre de votre page </TITLE>
</HEAD>
<BODY>
<H2 ALIGN=CENTER> Entete de votre page.</H2>
<P ALIGN=CENTER> <IMG SRC="picture.gif" ALT=Image WIDTH=80 HEIGHT=?
UNITS=pixels> </P>
<P> Mes <STRONG>sujets</STRONG> favoris
<OL>
<LI> Sujet 1
<LI> Sujet 2
<LI> Sujet 3
</OL>
<BR>
<HR>
<P> La liste de mes <EM>sites</EM> preferes
<UL>
<LI> Essayer ce <A HREF="http://adresse du site">site </A>
<LI> Un autre <A HREF="http://adresse du site">site</A> site que je
vous conseille !
<LI> Mon <A HREF="http://adresse du site">site</A> prefere
</UL>
<HR>
```

```
<P> Merci de votre visite  
<BR>  
Vous pouvez nous joindre a l'adresse : <ADDRESS><A  
HREF="MAILTO:"nom@user.com">adresse</A></ADDRESS>  
</BODY>  
</HTML>
```

3.B.3 Les types de pages

Les pages ou documents au format HTML peuvent être classés de la manière suivante (par complexité et coût croissants) :

- Les pages de texte : elles demandent un investissement minimum et l'apprentissage du langage HTML.
- Les pages graphiques : le niveau de qualité souhaité et la taille des fichiers peuvent nécessiter l'utilisation de logiciels (Paint Shop Pro par exemple) et des compétences graphiques. Ces éléments seront souvent sous-traités par des spécialistes de la conception graphique.
- Les pages liées à des bases de données : relèvent plus des connaissances de programmation et des SGDB Relationnels mais nécessitent également une étude ergonomique poussée.
- Les pages animées et multimédia sont les plus complexes à développer. Les coûts d'ingénierie et les temps de conception plus importants augmentent la complexité et le prix moyen de la page.

3.B.4 Relier le Web aux bases de données

Pour mettre en oeuvre des applications sur le Web et sur des Intranets, aménager un accès complet aux bases de données est indispensable. L'accès doit pouvoir s'effectuer en lecture mais aussi en mise à jour, dans un contexte où l'intégrité et la sécurité des données devront être respectées.

Sur les sites commerciaux, les pages doivent être personnalisées pour chaque client, afin que sa demande puisse obtenir une réponse appropriée. On entre donc dans une logique d'échanges dynamiques entre le Web et les bases de données de l'entreprise.

Pour cela, des interfaces existent, notamment la passerelle CGI, grande consommatrice de trafic par ses échanges continuels de données. Deux alternatives apparaissent les API et des passerelles propriétaires, économisant les ressources machines et les échanges

L'offre de nouveaux outils est en pleine explosion. De nombreux éditeurs de bases de données ou de serveurs Web proposent déjà des interfaces, mais beaucoup de logiciels sont attendus qui prendront en compte les standards émergents, comme Java.

Oracle propose un serveur HTTP nommé Oracle WebServer permettant une liaison aisée avec le SGBDR Oracle 7.

La démonstration d'Oracle WebServer : "Take the train"

Cette application permet d'effectuer une réservation de billets de train à partir d'un navigateur, les données ainsi que les pages étant stockées dans une base de données Oracle.

SOMMAIRE


Appel de la page
d'accueil de
l'application

Oracle Travel - Microsoft Internet Explorer

File Edit View Go Favorites Help

Address: <http://s00p133:90/demo/owa/tr.splash>

Welcome to Oracle Travel



Enjoy the comfort of travelling by train!

User:

Password:

Action:

Boîte de dialogue
pour la connexion à
la base de données
Oracle 7.

SOMMAIRE

Boîte de dialogue utilisée pour la réservation des billets

Preferences - Microsoft Internet Explorer

File Edit View Go Favorites Help

Address:

Name

Address

City

Zipcode

State

Country

Email

Balance \$

Language ☐ Deutsch ☐ English ☐ Español ☒ Français ☐ Nederlands ☐ Svenska

Page History ☐ Yes ☒ No

Comments about yourself

Résultat du processus de réservation d'un billet

fc Page principale - Microsoft Internet Explorer

File Edit View Go Favorites Help

Address: Refresh

fc, Bienvenue ; Oracle Travel sur le Web. Vous trouverez ici des informations sur ce fait lors de visites précédentes. Vous pouvez faire une nouvelle réservation en consultant [destinations possibles](#) Solde courant (en \$) :4857.5

Billets achetés

Etat	Destination	Date de départ	Date de retour	Nbre places	Prix
6 jours reste	Chicago	10-Jan-97	18-Jan-97	1	142.50

L'intranet peut devenir le noyau de toutes les applications.

On comprend l'intérêt de sites Web au contenu défini dynamiquement avec un accès aux bases de données, mais peut-on étendre ce type de technique à des applications classiques, dans le cadre d'un intranet ?

Aujourd'hui, le client-serveur est le modèle de pointe pour les applications autour des bases de données. Grâce aux progrès des "*middlewares*" destinés aux Web, l'intranet pourrait devenir à son tour la référence pour le développement des applications d'entreprises.

Le navigateur Web est déjà considéré comme un client universel pour accéder à tous types de ressources, traitements existants et bases de données.

Le serveur Web pour être considéré comme un serveur universel doté des interfaces nécessaires entre les demandes des clients et les serveurs du système.

Le gros avantage de l'approche "accès aux données" via le Web est d'éliminer toutes les liaisons préalablement nécessaires entre les clients et les serveurs, pour la rejeter entre les serveurs (du serveur HTTP au serveur de base de données par exemple).

Les runtimes de chaque environnement de développement (PowerBuilder, SQLWindows, etc.), installés sur chaque poste, devenus superflus, sont remplacés par le navigateur. Ce qui simplifiera, de surcroît, grandement le déploiement des applications.

3.B.5 Règles sur les pages HTML

Le développement de pages Web nécessite de se focaliser sur la satisfaction des utilisateurs. Dans ce souci, la présentation et la stylistique de vos pages sont primordiales.

a. Le contenu des pages

La première étape dans la conception d'une page consiste à déterminer le contenu de votre (ou vos) page(s). Ce contenu dépend bien sûr de l'objectif de votre site : Information ? Divertissement ? Education ? ... N'oubliez pas, les utilisateurs accèdent à votre site pour y trouver de l'information.

b. L'organisation et le style

Une page avec un bon contenu, ne sera pas consultée souvent, si l'information est désorganisée. Si vous n'avez pas d'idée précise pour l'organisation de vos pages, n'hésitez pas à consulter les pages disponibles sur les serveurs Web et inspirez-vous de l'organisation des pages qui vous plaisent.

Quelques conseils généraux peuvent cependant vous aider :

1. Mettez les informations les plus importantes dans la partie supérieure de la page (le titre de la page, l'objectif de la page, la date de la dernière mise à jour, le nombre d'accès à la page).

Adresse <http://www.hooked.net/~larrylin/web5.htm#1>

CREATING A SUCCESSFUL WEB PAGE



Last Updated 12-30-1996

21637 People have Visited this Site

2. Rédigez vos textes de façon claire et concise.
3. Ne surchargez pas vos pages d'images ou de graphiques. Les images augmentent, de façon sensible, le temps de chargement de la page.
4. Favorisez la lecture rapide de vos pages. Utilisez des sommaires et des listes.
5. Évitez le texte clignotant, certes attractif mais trop pénible. Préférez la mise en relief avec le caractère gras ou italique.
6. Le fond d'écran d'une page Web doit être sobre, rapide à charger et simple à gérer. Il est préférable d'utiliser une couleur unie, claire et sobre, comme d'ailleurs avec une interface graphique traditionnelle.

c. Quelques recommandations

Règle P1	Une page doit être informative et attractive
Règle P2	Une page ne doit pas être trop dépendante de certaines caractéristiques de HTML non supportées par tous les navigateurs.
Règle P3	La taille de la page doit être limitée (4 à 5Ko conseillé). Limiter la taille des graphiques et images entre 10 et 15Ko par page. La taille visuelle d'une page doit se limiter à 2 écrans 640*480.
Règle P4	Indiquez où des changements ont été effectués ainsi que la date de mise à jour de vos document
Règle P5	Vérifiez, régulièrement, la validité de vos liens
Règle P6	Autorisez les lecteurs de vos pages à vous adresser, par courrier électronique, leurs remarques et suggestions
Règle P7	N'utilisez pas les images des autres sans leur accord
Règle P8	Réserver l'usage des liens pour le renvoi à vos propres ressources. N'essayer pas de remplacer yahoo, en surchargeant vos pages de liens vers d'autres sites.

d. Exemple d'organisation de page d'accueil

La page d'accueil du site Web Sql Ingénierie illustre une organisation basée sur le découpage en plusieurs parties, grâce aux cadres ou "frames", et l'utilisation d'images.

Un bandeau de boutons d'action :
pour lancer des opérations sur la page active

[\[Accueil\]](#) - [\[Nouveautés\]](#) - [\[Mots clés\]](#) - [\[Projets\]](#) - [\[Formation\]](#) - [\[Produits\]](#) - [\[Etudes\]](#)



SQL Ingénierie

Le corps de page : qui contient les informations de la page

L'Actualité



[Le site du client serveur universel !](#)



[La revue de la presse](#)



Découvrez [notre société](#)



Le département [Etudes comparatives](#)

Une mine d'informations critiques sur les outils du marché. Issue des fameuses études comparatives menées par le département R&D de SQL Ingénierie.



Le département [Projets](#)

Notre engagement dans les projets client-serveur, Intranet et Datawarehouse. Une démarche pragmatique qui s'appuie sur les standards du marché, un savoir technique et ergonomique que vos utilisateurs apprécieront.

[Accueil](#)

[Nouveautés](#)

[Mots clés](#)

[Projets](#)

[Formation](#)

[Produits](#)

[Etudes](#)

Un bandeau de navigation, placé juste avant le pied de page, qui permet les changements de pages

Une Zone de raccourci (dans un "frame") pour passer de branche en branche dans l'application sans avoir à remonter toute l'arborescence. Les raccourcis visuels (image + texte) représentent les objets majeurs de l'application

Au 4 parties ci-dessus peuvent s'ajouter :

1. **L'entête** comportant le titre de la page et un texte explicatif de la page.
2. **Le pied de page** permettant d'identifier la page.

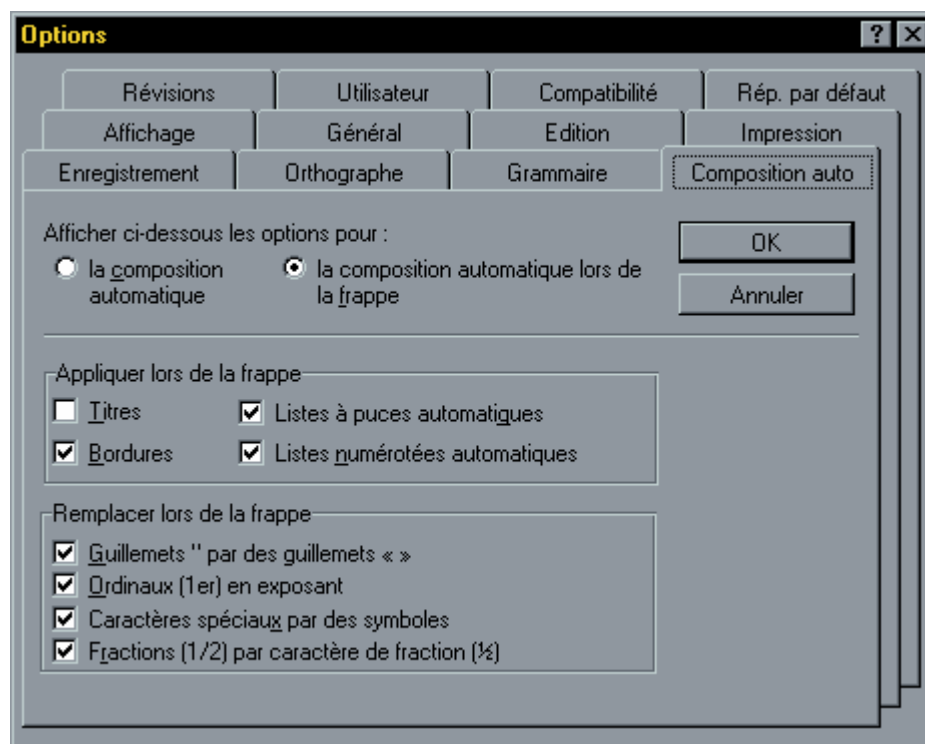
4. LA PROGRAMMATION ÉVÉNEMENTIELLE

La programmation événementielle permet de développer des applications pilotées par l'utilisateur et non plus des applications qui pilotent l'utilisateur.

L'application programmée de façon événementielle contient des procédures dites événementielles qui ne sont pas exécutées dans un ordre préétabli, mais qui s'exécutent à la demande pour réagir aux événements provoqués par la manipulation des objets graphiques de la part de l'utilisateur.

Cette façon de programmer est aujourd'hui la plus répandue. Notamment dans le environnement graphique comme Windows. Les langages ou les environnement de développement dits de quatrième génération (L4G) utilisent tous ce mode de fonctionnement : Access, Visual Basic, PowerBuilder, Oracle Power Objects, pour ne cités qu'eux, sont orientés objets et événements.

Dans la boîte de dialogue ci-contre, comptez le nombre d'actions que peut entreprendre l'utilisateur. Il y en a pas loin d'une trentaine : cliquer sur une case, déplacer la fenêtre ou changer d'onglet. Seule la programmation événementielle permet de prendre en charge un tel éventail de possibilités



4.A. QUELQUES PRINCIPES

- A part le fait que l'exécution des procédures est déclenchée par des événements, la programmation événementielle utilise les même principes et méthodes (algorithmique, structures de données, structures de contrôle) que la programmation procédurale traditionnelle.
- En programmation événementielle, l'application ne comporte pas forcément de programme principal. Les procédures de l'application ne sont appelées que si les événements qui leur correspondent sont engendrés.
- C'est le système d'exploitation par l'intermédiaire de son gestionnaire d'événements, lui-même attaché au gestionnaire de fenêtres, qui « indique » à l'application que l'événement a eu lieu.

SOMMAIRE

- La liste des événements programmables est prédéfinie et représente tout ou partie des événements pris en charge par le système d'exploitation et son interface graphique.
- C'est à l'application de prendre en charge les événements qu'elle désire traiter.
- Les environnements de développements qui permettent la programmation événementielle s'exécutent dans un environnement graphique et les composants de cette interface sont vus comme des objets au sens de la programmation orientée objets.
- Les événements sont affectés à des objets du système. Chaque objet « connaît » un certain nombre de propriétés qui permettent de définir son aspect visuel et/ou son comportement ainsi qu'un certain nombre d'événements auxquels le développeur doit associer des procédures pour faire réagir l'objet.
- Si pour un événement donné, l'application doit réagir, elle déclenche une ou plusieurs actions qui vont modifier l'état des objets qu'elle gère.

4.B. LES DIFFERENTS TYPES D'ÉVÉNEMENTS

4.B.1 Événements externes

Ce sont les événements liés à une action de l'utilisateur. Par exemple : clic, déplacement de souris, pression sur une touche de clavier, relâchement d'une touche, etc.

4.B.2 Événements internes

Ce sont des événements liés à un changement d'état de l'application à l'issue de l'exécution d'une opération. Par exemple : ouverture d'une fenêtre, écriture d'un enregistrement dans une base de données, écoulement d'un certain délai, fin de l'application.

4.B.3 Événements inter-applications

Ils permettent à des applications d'échanger des informations ou des données.

Par exemple, dans l'environnement Windows, les protocoles DDE (*Dynamic Data Exchange*) et OLE (*Object Linking and Embedding*) permettent, par l'échange d'événements, d'établir un lien entre une application cliente et une application serveur et d'échanger des informations.

4.C. LES OBJETS GRAPHIQUES

Un objet graphique est un outil élémentaire de dialogue avec l'utilisateur. Par exemple : une case à cocher, un bouton de commande, un menu, etc. Les objets graphiques sont définis au niveau de l'interface graphique du système d'exploitation, toutes les applications les utilisent. C'est bien parce qu'il sont identiques d'une application à l'autre que l'interface est homogène, cohérente et que les applications sont faciles à « découvrir ».

Du point de vue du développeur, un objet graphique est une référence à un objet au sens de la programmation orientée objets.

On parle aussi de « contrôle » pour désigner un objet graphique dans la mesure où celui-ci contrôle le dialogue avec l'utilisateur.

4.C.1 Définition des objets graphiques

Les objets graphiques possèdent :

1. Une représentation visuelle précise qui permet à l'utilisateur d'identifier leur rôle immédiatement
2. Des propriétés qui sont autant de variables auxquelles il est possible d'affecter des valeurs pour définir les paramètres de fonctionnement de l'objet. Par exemple : la propriété *Titre* pour définir le texte affiché au centre d'un bouton de commande, la propriété *Cochée* pour modifier l'état d'une case à cocher (oui ou non).
3. Des méthodes qui sont autant de procédures prédéfinies qui sont applicable à l'objet pour le faire réagir. Par exemple : la méthode *Montre* pour afficher une fenêtre, la méthode *Affiche* pour afficher quelque chose dans une fenêtre.

4.C.2 Manipulation des objets graphiques

a. Le focus

L'objet qui a le focus est celui sur lequel va s'appliquer la prochaine action de l'utilisateur. L'objet qui a le focus se voit « attribuer » le clavier et la souris. Dans Windows, par exemple, on donne le focus en cliquant avec la souris ou en utilisant la touche Tabulation.

b. Sélection

La sélection permet de désigner l'objet ou le groupe d'objets sur lequel va s'appliquer la prochaine action. Il existe différents types de sélection : simple (un seul élément), multiple contiguë (plusieurs éléments les uns à côté des autres), multiple disjointe (plusieurs éléments séparés) ou hétérogène (plusieurs éléments de différentes natures).

c. Navigation

La navigation grâce au clavier ou à la souris permet faire passer le focus d'un objet graphique à un autre.

d. Manipulation directe

C'est la transcription métaphorique de la manipulation des objets sur le bureau par « Glisser-Déplacer ». Par exemple : supprimer un en le déplaçant de l'explorateur vers la corbeille

e. Le retour d'informations (*feed-back*)

Le *feed-back* est l'information que fournit l'application à son utilisateur pour lui indiquer ce qui est en train de se passer. Par exemple : le pointeur de souris devient un sablier pour indiquer à l'utilisateur qu'il doit patienter. Quand on installe une nouvelle application, une barre de progression montre en permanence la répartition entre ce qui est fait et ce qui reste à faire : l'utilisateur sait toujours où il en est.

5. INTERFACE GRAPHIQUE ET PROGRAMMATION EVENEMENTIELLE AVEC ACCESS ET VISUAL BASIC

Dans la base Comptoirs, livrée avec Access 7, intéressons-nous au formulaire intitulé « Dialogue Etats ventes ». Ce formulaire représente une fenêtre Windows qui sert de boîte de dialogue et permettant à l'utilisateur de choisir le document qu'il souhaite imprimer.

Avec une interface en mode caractères sur un terminal de 25 lignes de 80 caractères, le dialogue pourrait être le suivant :

Une interface en mode caractères. L'utilisateur est guidé pas à pas, il ne peut pas faire d'autres choix que ceux qui lui sont proposés. Il ne peut pas facilement revenir sur l'un de ses choix ou interrompre son travail.

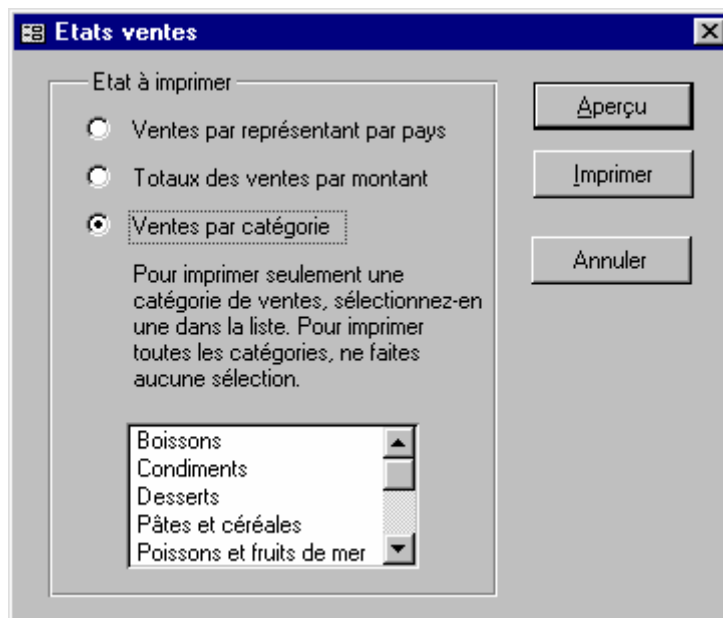
```
Que voulez-vous imprimer ?
1 - Les ventes par représentants par pays
2 - Les totaux des ventes par montant
3 - Les ventes par catégorie
Votre choix :

/* si choix 3 */
Quelle catégorie voulez-vous imprimer ?
1 - Boissons
...
n - Viandes
Votre choix :

Souhaitez-vous :
1 - Un aperçu sur écran
2 - Une impression sur papier
Votre choix :
```

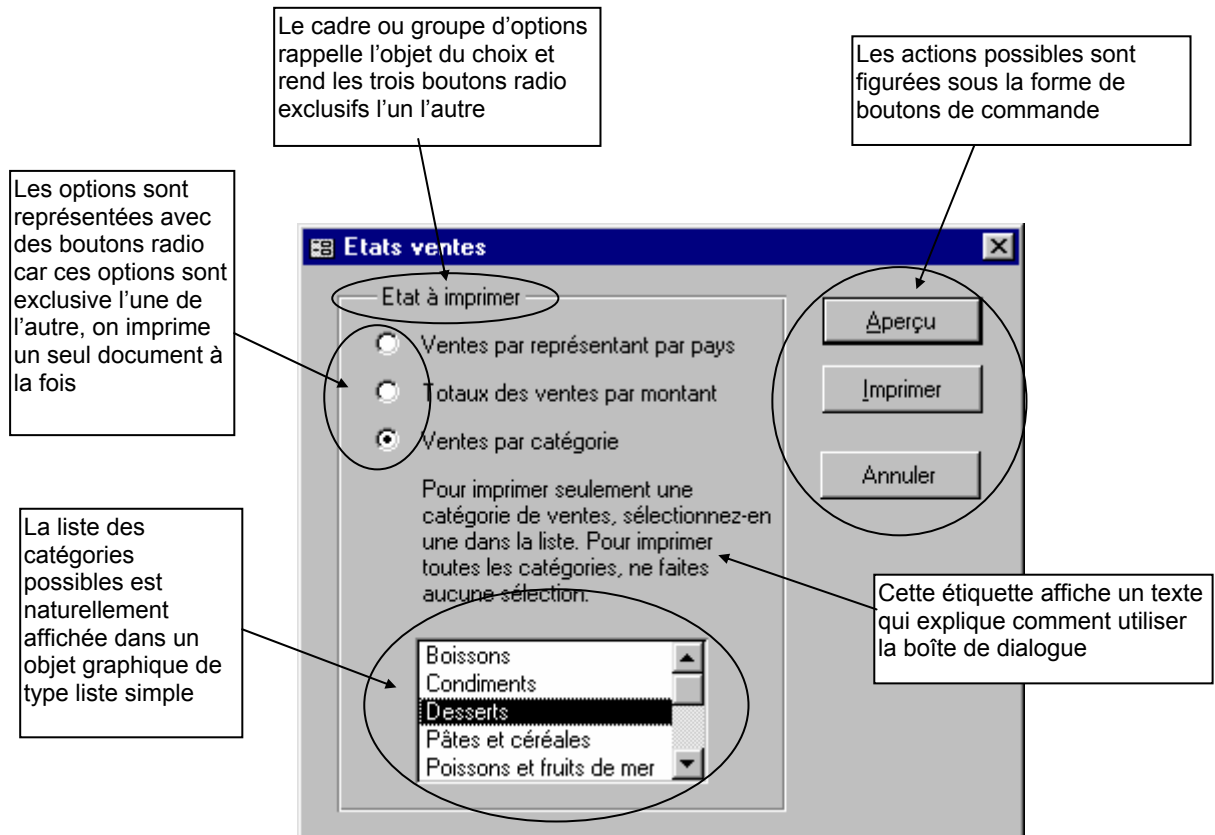
Voici le même dialogue en mode graphique tel que le propose Access sous Windows 95 :

Une interface en mode graphique. Toutes les possibilités de choix sont visibles, c'est l'utilisateur qui est aux commandes. Il peut revenir à tout moment sur l'un de ses choix.

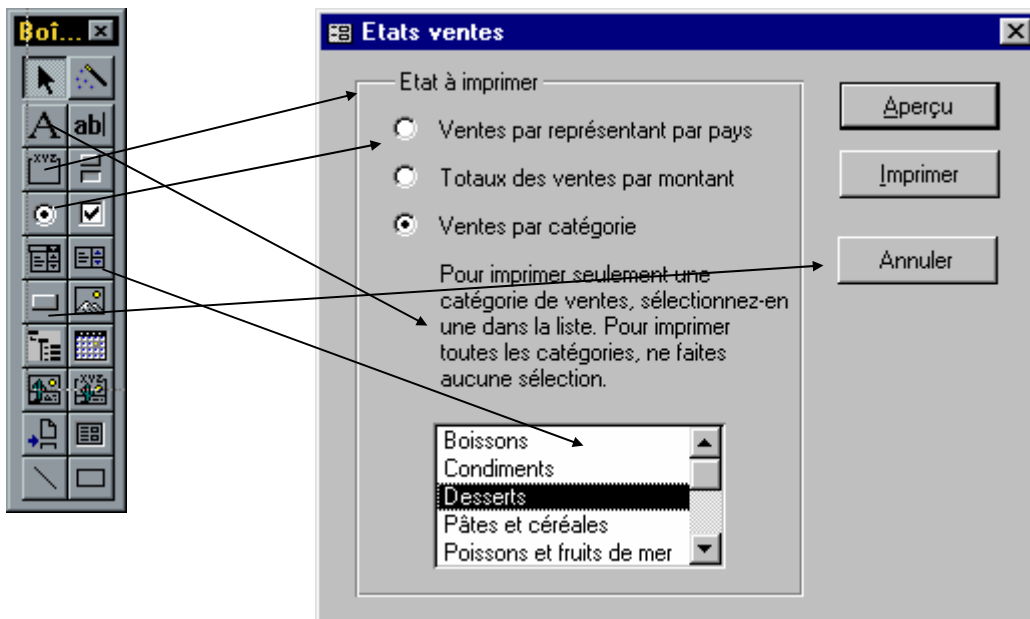


La liste des catégories n'est active que lorsque l'utilisateur clic sur le bouton radio « Ventes par catégorie ».

5.A. Organisation de l'interface



Au moment de la création de la boîte de dialogue, une boîte à outils (une barre d'outils) propose l'ensemble des objets graphiques (ou contrôles) disponibles :



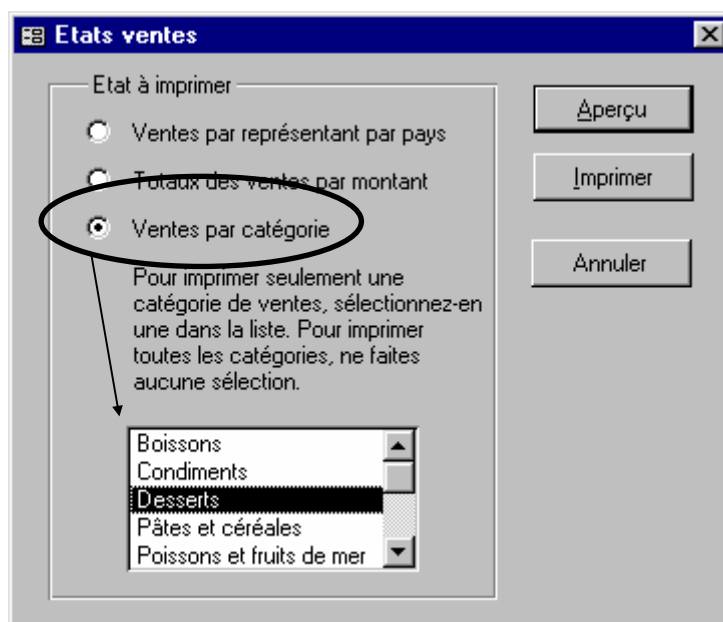
5.B. Programmation événementielle de l'interface

5.B.1 Activer la liste des catégories

Intéressons-nous, par exemple, à la façon de rendre disponible la liste des catégories au moment où l'utilisateur choisit la troisième option (ventes par catégorie). Si l'utilisateur opère l'un des deux premiers choix, cette liste est rendue inactive : elle est visible mais n'est pas utilisable (elle ne peut pas prendre le focus).

Le fait de cliquer sur le troisième bouton radio (ou bien d'appuyer deux fois sur la touche flèche vers le bas avec le clavier) fait réagir l'application : elle doit rendre active la liste des catégories.

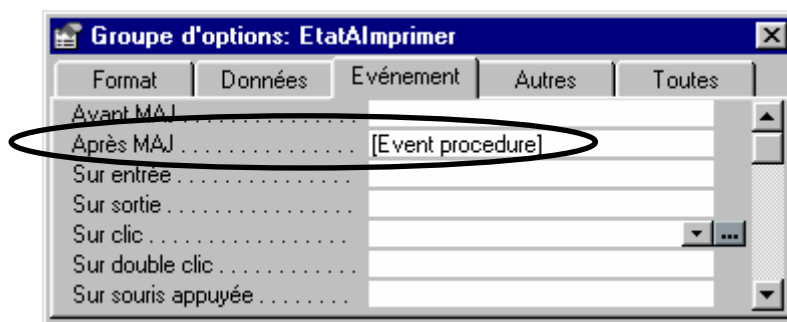
Quand le troisième bouton radio est sélectionné, l'objet graphique (le contrôle) liste est activé et devient utilisable



Dans Access, le groupe d'options prend la valeur du bouton radio sélectionné qu'il contient, à savoir ici les valeurs 1, 2 ou 3. Le groupe d'options « Etat à imprimer » est un objet graphique nommé « EtatAlmprimer » dans l'application. Quand l'utilisateur sélectionne un bouton radio, la valeur de l'objet graphique EtatAlmprimer change. Par exemple, quand l'utilisateur sélectionne le bouton radio « Vente par catégorie », la valeur du contrôle EtatAlmprimer change, il prend la valeur 3.

Le fait que la valeur d'un groupe d'options change est un événement détecté par le système d'exploitation. Dans Access cet événement s'appelle « Après mise à jour » (Après MAJ).

Parmi les propriétés du groupe d'options « EtatAlmprimer », on trouve des propriétés d'événements qui permettent d'associer une procédure événementielle (*Event procedure*) à chaque événement possible sur cet objet.



La procédure associée est rédigée en langage Visual Basic :

```
Private Sub EtatAlmprimer_AfterUpdate()  
' Active la zone de liste modifiable SélectionCatégorie si  
l'utilisateur a choisi l'état Ventes par catégorie.  
  
    Const conVentesParCatégorie = 3  
  
    If Me!EtatAlmprimer.Value = conVentesParCatégorie Then  
        Me!SélectionCatégorie.Enabled = True  
    Else  
        Me!SélectionCatégorie.Enabled = False  
    End If  
  
End Sub
```

Les deux premiers mots *Private Sub* indique que nous avons à faire à une procédure (*Sub routine*) locale (privée ou *Private*) à la boîte de dialogue (utilisable uniquement dans le contexte de cette boîte de dialogue).

Remarquez le nom de la procédure : `EtatAlmprimer_AfterUpdate()`. « `EtatAlmprimer` » est le nom du contrôle concerné et « `AfterUpdate` » est le nom anglais de l'événement `Après MAJ`. Il s'agit de la procédure qui s'exécute lorsque l'événement « `Après mise à jour` » a lieu sur le groupe d'options « `Etats à imprimer` ».

La constante « `conVentesParCatégorie` » prend la valeur 3.

La syntaxe « `Me!EtatAlmprimer.Value` » signifie : propriété *Valeur* (*Value*) de l'objet « `EtatAlmprimer` » contenu dans le formulaire en cours (*Me*).

Le test (*If*) vérifie si le groupe d'options a pris la valeur de l'option 3. Si oui, on active la liste des catégories de la façon suivante :

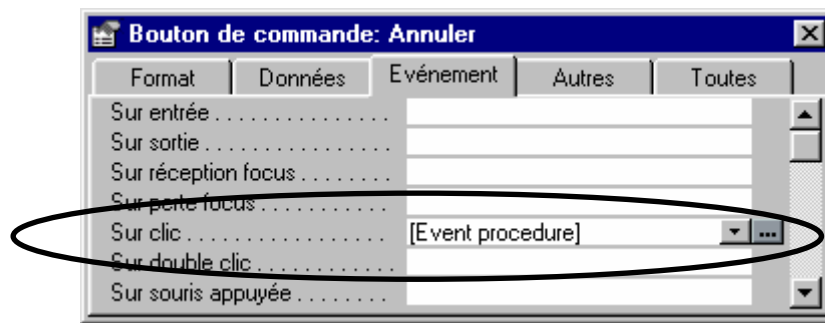
La syntaxe « `Me!SélectionCatégorie.Enabled = True` » affecte la valeur *Vrai* (*True*) à la propriété *Activé* (*Enabled*) de l'objet « `SélectionCatégorie` » contenu dans le formulaire en cours (*Me*).

5.B.2 Annuler le travail en cours

Que se passe t-il quand l'utilisateur renonce à imprimer un document et clique sur le bouton Annuler afin de fermer la boîte de dialogue ?

Ce bouton s'appelle fort logiquement « *Annuler* ».

Une procédure événementielle est associée à l'événement « *Sur clic* » (en anglais *Click*) :

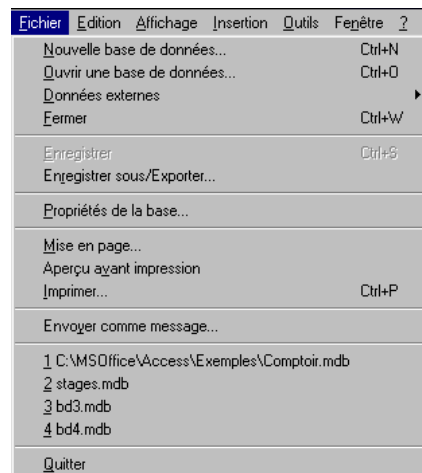


Cette procédure ferme tout simplement la fenêtre de la boîte de dialogue :

```
Private Sub Annuler_Click()  
...  
' Ferme le formulaire.  
    DoCmd.Close  
...  
End Sub
```

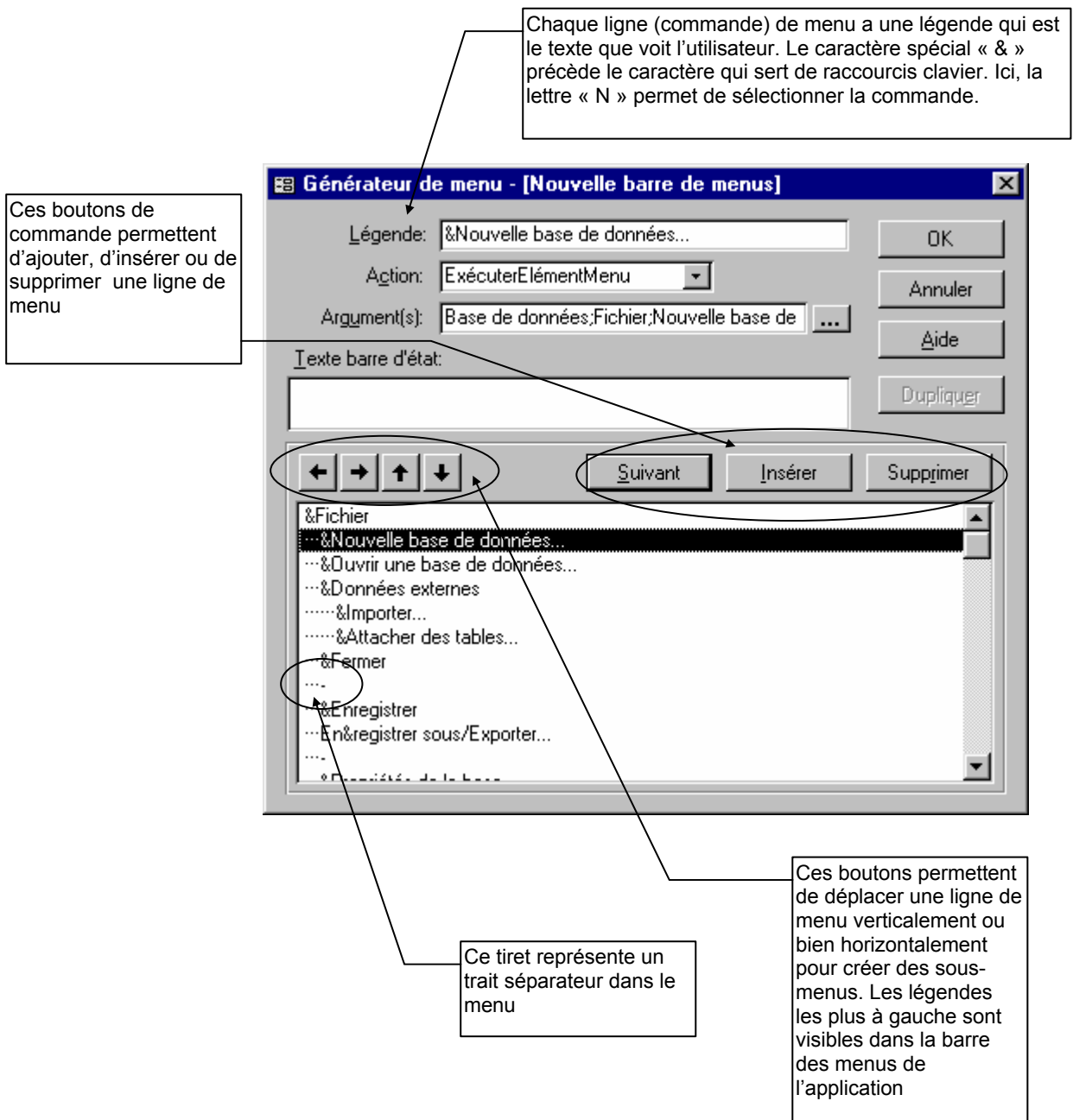
5.B.3 Création de menus

La création d'un menu dans les environnement Access ou Visual Basic est assez simple. Observons le menu menu Fichier d'Access :



SOMMAIRE

Ce menu peut être obtenu avec le générateur de menus.



Module 2 **Système de gestion de base de données relationnel**

1. PRÉSENTATION

Ce module présente la notion de système de gestion de bases de données.

Il s'agit pour l'essentiel de comprendre le rôle d'un tel outil logiciel et d'être capable d'en utiliser les possibilités. Dans l'optique de la profession, il s'agit de savoir expliquer les principes de fonctionnement d'un tel système et d'être capable d'en utiliser les commandes.

1.A. Références techniques

Ce cours s'appuie pour l'essentiel sur deux références incontournables du marché des bases de données : Microsoft Access pour l'environnement Windows et Oracle en tant que leader de ce marché.

Ce cours ne constitue pas un mode d'emploi de ces logiciels, pour apprendre à utiliser Microsoft Access ou le SGBD Oracle, utilisez la documentation en ligne et sur papier livrée avec le logiciel ou/et consultez les références bibliographiques fournies ci-dessous.

Il est bien entendu vivement conseillé de pratiquer Microsoft Access et/ou Oracle pour assimiler ce cours de façon satisfaisante. Afin de vous permettre de retrouver facilement comment les copies d'écran du cours ont été réalisées avec Access, vous trouverez dans la marge comment procéder pour voir la même chose avec Access.

La notation utilisée est du type « Nom de menu / Nom d'une commande de ce menu / nom d'une option, etc... ». Le signe BDD signifie « fenêtre base de données » et indique que l'on part de la fenêtre base de donnée.

1.B. Références bibliographiques

Il existe de très nombreux ouvrages sur Microsoft Access. Ils sont disponibles dans toute la gamme des prix du livre de poche jusqu'à la « bible » plus complète et accompagnée d'un cédérom. Il faut savoir que la documentation « en ligne » ainsi que la documentation sur papier de ce logiciel est également très bien conçue.

- Pour une initiation pas à pas : Microsoft Access par étapes chez Microsoft Press.
- Pour une vue d'ensemble en livre de poche : Access 95 chez Micro Application (MA) collection PC Poche.
- Pour aller plus loin : Microsoft Access au quotidien chez Microsoft Press.
- Pour une vue d'ensemble du SGBDR Oracle : Oracle 7 - Langages, architecture, administration chez Eyrolles.

2. NOTION DE BASE DE DONNÉES

2.A. Qu'est-ce qu'un SGBDR ?

Une base de données est un ensemble cohérent d'informations mémorisées sur support informatique. Ces informations sont accessibles à l'aide d'une application appelée système de gestion de base de données (SGBD). Si ce SGBD est basé sur le modèle relationnel de CODD, on dit qu'il s'agit d'un système de gestion de base de données relationnel (SGBDR). Pour dialoguer avec un SGBDR on utilise le langage SQL. Ce langage permet de soumettre des requêtes (des questions) au SGBDR.

Un Système de Gestion de Base de Données est un ensemble de logiciels qui fournit un environnement pour :

- décrire
- mémoriser
- manipuler
- traiter des collections de données

tout en assurant leur

- sécurité
- confidentialité
- intégrité

alors qu'un grand nombre d'utilisateurs variés interagit avec ces collections de données.

2.B. Caractéristiques

Nous allons énoncer de manière précise les caractéristiques d'un Système de Gestion de Bases de Données.

2.B.1 Indépendance physique

Il s'agit d'obtenir l'indépendance de la structure de stockage des données physiques par rapport à la structure des données du monde réel (sa représentation conceptuelle).

2.B.2 Indépendance logique

Il s'agit de permettre une certaine indépendance entre la description des données vues par une application ou un groupe de travail et la structure des données dans le monde réel.

2.B.3 Efficacité des accès aux données

Mesurée essentiellement en termes de nombre d'accès disques.

2.B.4 Administration cohérente des données

Cela concerne le contrôle des données, la résolution des conflits d'accès, l'optimisation des accès aux données et l'utilisation du matériel. Une administration centralisée des données est souvent plus cohérente et efficace.

2.B.5 Non redondance des données

Ce qui permet d'éviter les mises à jour multiples, les incohérences et la perte de place en mémoire.

2.B.6 Cohérence des données

A chaque modification des données, leur cohérence doit être vérifiée par le système à l'aide des contraintes d'intégrité qui définissent cette cohérence.

2.B.7 Partageabilité des données

Permettre à plusieurs applications ou utilisateurs de partager simultanément les données de la base, tout en leurs donnant l'impression d'être seuls à les utiliser.

2.B.8 Sécurité des données

Protection contre les accès non autorisés, contre les pannes, etc... Dans la pratique, ces objectifs, dont certains sont antinomiques, ne sont que partiellement atteints.

Par exemple, l'efficacité des accès aux données est peu compatible avec l'indépendance physique.

3. ÉTAT DU MARCHÉ DES SGBDR

3.A. Les générations de SGBD

Un Système de Gestion de Bases de Données est aujourd'hui un logiciel de base essentiel dans un système informatique de gestion.

L'histoire des SGBD peut être résumée en distinguant trois générations qui correspondent à différents modèles de données.

3.A.1 Les SGBD basés sur les modèles d'accès

La première génération de SGBD s'appuie sur des modèles de données hiérarchiques et réseaux. Ces SGBD basés sur les modèles d'accès tendent à privilégier l'optimisation des entrées/sorties.

Elle vise à étendre un système de gestion de fichiers (l'ancêtre des SGBD) par des possibilités de liaisons interfichiers matérialisés par des pointeurs. Les bases de données sont représentées au niveau des types d'articles par une hiérarchie ou un graphe. Un SGBD de première génération fournit une faible indépendance physique compliquant ainsi l'administration et la manipulation des données.

En particulier, son langage de manipulation des données navigationnel impose au programmeur de spécifier les chemins d'accès aux données en naviguant dans le graphe de la base.

Ces SGBD sont historiquement les plus anciens, mais sont encore répandus, notamment dans la "grande" informatique, pour des raisons de performances, de fiabilité, de taille du parc des systèmes installés, et d'importance des applications construites autour d'eux.

3.A.2 Les SGBD basés sur le modèle relationnel

La deuxième génération de SGBD est née vers 1970 avec l'apparition du modèle de données relationnel.

Une dizaine d'années d'efforts de recherche et de développement furent nécessaires pour aboutir à la commercialisation des premiers SGBD relationnels. L'accent est mis sur la facilité de la représentation et de l'utilisation des données. Cette facilité est obtenue au détriment des performances. Dans ces SGBD, les données sont représentées sous la forme de tableaux qui peuvent être mis en correspondance par l'intermédiaire de clés communes.

Depuis, la technologie des bases de données relationnelles a fait l'objet de progrès remarquables en termes de facilité d'usage et de performances. Aujourd'hui tout SGBD Relationnel offre un ensemble intégré d'outils basés sur un langage de 4ème génération (L4G) afin d'accroître la productivité des utilisateurs.

Les premières et les deuxièmes générations de SGBD ont été conçues pour les applications de gestion classique, comme par exemple, la gestion des stocks ou la comptabilité d'une entreprise.

Deux tendances complémentaires récentes soulignent les limites de ces systèmes.

La première est la manifestation de besoins pressants en gestion des données de la part d'applications nouvelles, telle que la bureautique, la conception assistée par ordinateur ou le génie logiciel, qui se singularisent par rapport aux applications traditionnelles de gestion.

La seconde tendance est la diversité croissante des environnements informatiques, essentiellement décentralisés, qui favorisent la répartition et l'hétérogénéité des données.

3.A.3 Les SGBD Avancés

On parle souvent de **troisième génération** de SGBD pour désigner des systèmes qui supportent bien ces applications nouvelles ou exploitent des environnements opérationnels complexes (répartis, parallèles ou hétérogènes). Cette nouvelle génération de SGBD, souvent qualifiés d'avancés par opposition aux précédents, peut être étudiée par trois types essentiels qui ont pour ambition de pallier les limites des SGBD relationnels.

Ce sont :

- les **SGBD Orientés Objets**,
- les **SGBD Déductifs** et
- les **SGBD Répartis**.

Les bases de données orientées objets ont pour objectif essentiel le support d'objets complexes et dynamiques.

Cette approche consiste à intégrer les techniques des langages de programmation orientés objets (comme C++) et des bases de données afin d'offrir un langage unique et général pour programmer les applications traditionnelles et récentes des bases de données. La conception de la base de données repose sur l'utilisation d'un modèle de données objet.

Certains SGBD Relationnels proposent des solutions pertinentes en matière de répartition de données.

3.B. Les SGBD Relationnels

Les SGBD Relationnels représentent plus de 80 % du total des ventes de systèmes de gestion de bases de données. Leurs atouts sont :

- la normalisation du langage d'interrogation,
- la présence d'outils d'amélioration des performances,
- la stabilité des fournisseurs.

Pour les applications très pointues, notamment dans le domaine du transactionnel, il existe des machines dédiées : les serveurs de bases de données, voire les machines massivement parallèles.

3.B.1 SGBD Relationnels sur gros systèmes

Ces SGBD disposent en plus des fonctions classiques des SGBD, d'un langage de 4^{ème} génération destiné à permettre l'utilisation du SGBD par des non-informaticiens.

Société Productrice	SGBD	L4G
IBM	DB2 SQL/DS	SQL/DBE
COMPUTER ASSOCIATES	CA-UNIVERSE CA-DB CA-DATACOM	
BULL/HONEYWELL	MRDS	
ORACLE CORPORATION	ORACLE (V7)	Sql*Forms puis Developer/2000
INFORMIX	INFORMIX ONLINE	I4GL
ASK INGRES	INGRES	Windows 4GL
SYBASE	SQL-SERVER	Powerbuilder
PROGRESS SOFTWARE	PROGRESS	PROGRESS 4GL

3.B.2 SGBD Relationnels sur micro-ordinateur

Ce type de SGBD a connu un très grand succès ces dernières années, du fait de sa grande simplicité d'utilisation.

Pour certains d'entre eux, Oracle en particulier, il existe à la fois des versions pour micro-ordinateurs (Personal Oracle 7 pour Windows 3.1 et Windows 95), des serveurs départementaux (Oracle 7 Workgroup Server) et pour mini et gros systèmes (Oracle 7 Universal Server).

a. SGBD Relationnels Serveurs sur micro-ordinateur

Un SGBDR Serveur doit permettre à plusieurs d'utilisateurs d'accéder simultanément aux données d'une base de données relationnelle. Ceci n'est possible que si le SGBDR gère correctement les problèmes de concurrences d'accès et fournit des mécanismes de validation et d'annulation de transactions, pour garantir, dans toutes les situations, la cohérence de la base de données.

Les architectures serveurs SQL pour la plate-forme Intel commencent à avoir la faveur des entreprises. En effet, avec le coût toujours plus bas des processeurs, des dispositifs de stockage et de la mémoire, les solutions sur micro-ordinateurs deviennent crédibles et surtout compétitives.

De plus les systèmes d'exploitation 32 bits tels que Windows NT (Microsoft), Netware (Novell) ou OS/2 (IBM) et sont bien adaptés à ces applications : leurs atouts s'appellent performances, sécurité et relative facilité de mise en oeuvre.

Les architectures Client/Serveur SQL sont censées gérer des bases de données volumineuses (plusieurs giga octets de données), accessibles à un grand nombre d'utilisateurs : c'était naguère l'apanage des mini et grands systèmes , voire des stations multiprocesseurs en environnement Unix.

La technologie Client/serveur, elle, décharge le frontal client de tous les traitements SQL qui seront effectués par le serveur : de la sorte, seuls les résultats transitent par le réseau, réduisant d'autant le trafic.

Parmi les produits les plus représentatifs du marché nous citons :

- Oracle
- SQL Server de Microsoft
- Sybase
- Ingres Server
- OnLine d'Informix.

b. SGBD Relationnels Clients sur micro-ordinateur

Ces SGBDR sont essentiellement prévus pour fonctionner dans un environnement mono-utilisateur. De ce fait, ils ne disposent pas toujours de mécanismes pour contrôler les accès concurrents et n'offrent pas toujours la prise en charge des transactions. Cependant, avec des "extensions réseaux", ils sont parfois utilisés, pour des applications avec un nombre d'accès limité et des volumes réduits, dans un contexte multi-utilisateurs.

Voici quelques références de SGBD micros:

DBASE III,IV,V : de Ashton Tate - Borland

Dbase III : SGBDR de 1ère génération

Dbase IV : SGBDR de 2ème génération

Dbase V version Windows de Dbase 4

PARADOX 4.5 à 7 de Borland : SGBDR de 2ème génération

Paradox est un gestionnaire complet de Bases de Données qui vise un public varié allant de l'utilisateur à l'administrateur de base de données.

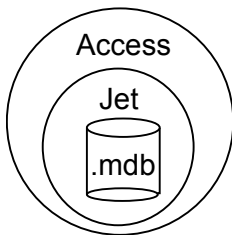
Microsoft Access 1.0 (oct 92)

Ms/Access 2.0 (mai 94)

Ms/Access 7.0 pour Windows 95 (sept 95)

4. UN SGBDR : MICROSOFT ACCESS

Le système de gestion de bases de données Access est devenu relativement incontournable dans l'environnement Windows. C'est un outil simple d'emploi qui peut revendiquer la plupart des caractéristiques d'un SGBDR tout en conservant l'ergonomie des applications Windows. Le développement d'applications sophistiquée reste cependant assez complexe, une bonne maîtrise du langage Visual Basic (Access Basic pour la version 2) est nécessaire.

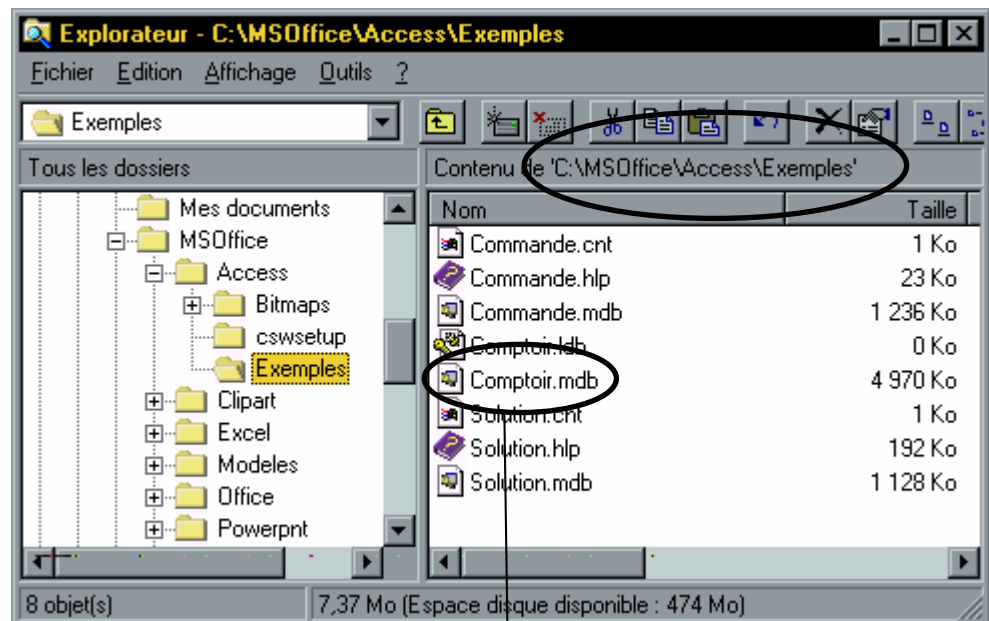


Access est un environnement de développement qui permet de créer des applications qui utilisent une base de données. La base de données elle-même est gérée par un moteur de base de données indépendant qui s'appelle Jet. On appelle « moteur », la couche logicielle qui gère effectivement les données en interprétant les requêtes en provenance d'une application. Le moteur Jet est également livré avec l'environnement de développement Visual Basic.

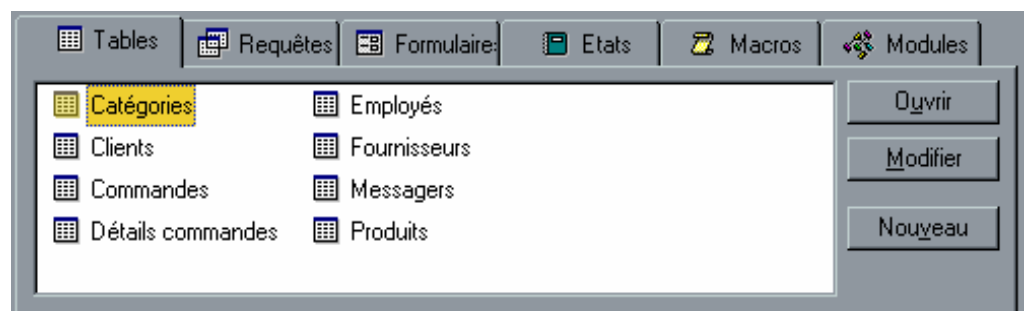
4.A. Le stockage et l'accès aux données

Access crée un fichier qui porte l'extension MDB (*Microsoft DataBase*) pour chaque base de données, par exemple « biblio.mdb » ou « comptoir.mdb ».

Position la plus courante de la base comptoir.mdb



La fenêtre Base de données (BDD) est accessible rapidement avec la touche F11

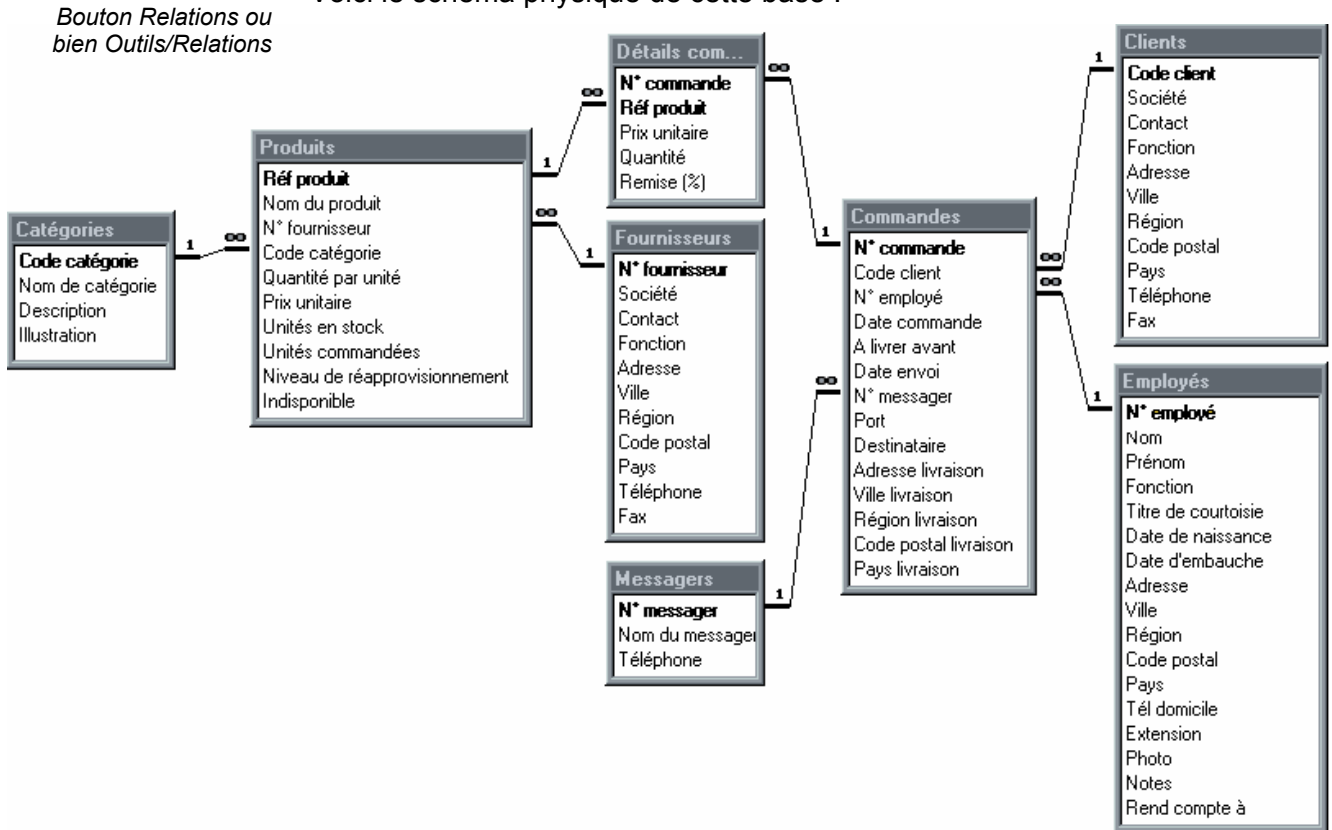


4.B. Le schéma physique d'une base de données

Parmi les trois niveaux d'abstraction pour l'analyse des données (conceptuel, logique, physique), une base de données se situe au niveau physique. On parle de schéma physique pour désigner la représentation de l'organisation des données dans une base de données.

Dans la suite de ce cours nous utiliserons la base « COMPTOIR.MDB » qui est livrée avec Access.

Voici le schéma physique de cette base :



4.B.1 Tables

L'unité de stockage dans une base de données relationnelle est la table. Une table regroupe un ensemble de données qui décrivent un même objet. Chaque cadre dans le schéma physique ci-dessus représente une table.

Clients
Code client
Société
Contact
Fonction
Adresse
Ville
Région
Code postal
Pays
Téléphone
Fax

Voici la table Clients qui contient les informations qui décrivent les clients de la société « Les Comptoirs ». On note que la donnée Code client est notée en gras, ceci signifie que cette donnée permet d'identifier chaque client. On dit aussi que « Code client » est la clé primaire de la table Clients.

La donnée Société est le nom de la société cliente.

Produits
Réf produit
Nom du produit
N° fournisseur
Code catégorie
Quantité par unité
Prix unitaire
Unités en stock
Unités commandées
Niveau de réapprovisionnement
Indisponible

Voici la table Produits qui contient les données qui décrivent les produits qui sont au catalogue de la société.

Détaillons le contenu de cette table :

Produits : Table

	Nom du champ	Type de données	Description
?	Réf produit	NuméroAuto	Numéro automatiquement assigné aux nouveaux produits.
▶	Nom du produit	Texte	
	N° fournisseur	Numérique	Identique au N° fournisseur de la table Fournisseurs.
	Code catégorie	Numérique	Identique au Code catégorie de la table Catégories.
	Quantité par unité	Texte	(par exemple : bouteille 1 litre x 12).
	Prix unitaire	Monétaire	
	Unités en stock	Numérique	
	Unités commandées	Numérique	
	Niveau de réapprovisionnement	Numérique	Nombre minimum d'unités à garder en stock.
	Indisponible	Oui/Non	'Oui' signifie que le produit n'est plus disponible.

Propriétés du champ

Général

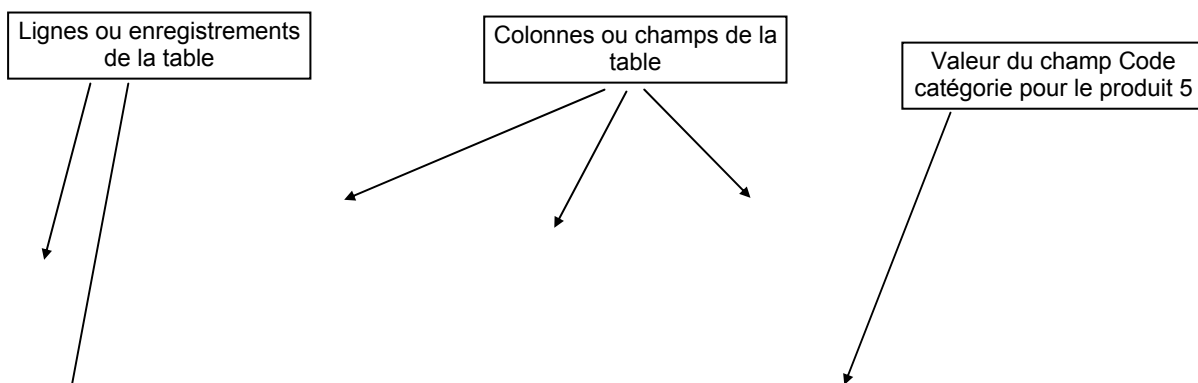
Liste de choix

Taille du champ	40
Format	
Masque de saisie	
Légende	Nom du produit
Valeur par défaut	
Valide si	
Message si erreur	
Null interdit	Oui
Chaîne vide autorisée	Non
Indexé	Oui - Avec doublons

Un nom de champ peut compter jusqu'à 64 caractères, espaces inclus. Pour obtenir de l'aide, appuyez sur F1.

A chaque donnée est associé un certain type ainsi que certaines propriétés qui décrivent la donnée.

Voici maintenant un extrait de la liste des valeurs effectivement présentes dans la table, chaque colonne représente une donnée, chaque ligne décrit un produit :



Réf produit	Nom du produit	N° fournisseur	Code catégorie	Quantité par unité	Prix unitaire
1	Chai	1	1	10 boîtes x 20 sacs	90,00 F
2	Chang	1	1	24 bouteilles (1 litre)	95,00 F
3	Aniseed Syrup	1	2	12 bouteilles (550 ml)	50,00 F
4	Chef Anton's Cajun Seasoning	2	2	48 pots (6 onces)	110,00 F
5	Chef Anton's Gumbo Mix	2	2	36 boîtes	106,75 F
6	Grandma's Boysenberry Spread	3	2	12 pots (8 onces)	125,00 F
7	Uncle Bob's Organic Dried Pears	3	7	12 cartons (1 kg)	150,00 F
8	Northwoods Cranberry Sauce	3	2	12 pots (12 onces)	200,00 F
9	Mishi Kobe Niku	4	6	18 cartons (500 g)	485,00 F
10	Ikura	4	8	12 pots (200 g)	155,00 F
11	Queso Cabrales	5	4	1 carton (1 kg)	105,00 F
12	Queso Manchego La Pastora	5	4	10 cartons (500 g)	190,00 F

Enr: 1 2 3 4 5 6 7 8 9 10 11 12 sur 77

BDD / Tables /
Produits / Ouvrir

Nombre de lignes dans la
table

4.B.2 Index

Un index permet d'accélérer la recherche des lignes d'une table sur une ou plusieurs valeurs de colonne. La clé primaire est le plus souvent indexée car sa valeur est souvent utilisée pour retrouver une ligne. D'autres données peuvent être indexées dès qu'elles sont fréquemment utilisées pour sélectionner certaines lignes d'une table.

Un index peut admettre des doublons ou non. La clé primaire n'admet pas de doublons car sa valeur est nécessairement différente pour chaque ligne du fait qu'elle permet d'identifier la ligne. Si l'index admet des doublons, plusieurs lignes de la table peuvent contenir des valeurs d'index identique.

Par exemple, dans la table Produits ci-dessus, la colonne Réf produit est un index sans doublon alors que le numéro fournisseur est un index qui autorise les doublons (différentes lignes ont le même numéro de fournisseur).

?	Nom du champ	Type de données	Description
	Réf produit	NuméroAuto	Numéro automatiquement assigné aux nouveaux produits.
	Nom du produit	Texte	
	N° fournisseur	Numérique	Identique au N° fournisseur de la table Fournisseurs.
	Code catégorie	Numérique	Identique au Code catégorie de la table Catégories.
	Quantité par unité	Texte	(par exemple : bouteille 1 litre x 12).
	Prix unitaire	Monétaire	
	Unités en stock	Numérique	
	Unités commandées	Numérique	
	Niveau de réapprovisionnement	Numérique	Nombre minimum d'unités à garder en stock.
	Indisponible	Oui/Non	'Oui' signifie que le produit n'est plus disponible.

Propriétés du champ

Général

Taille du champ : Entier long

Format :

Décimales : Auto

Masque de saisie :

Légende : N° fournisseur

Valeur par défaut : GenUniqueID()

Valide si :

Message si erreur :

Null interdit : Non

Indexé : **Oui - Avec doublons**

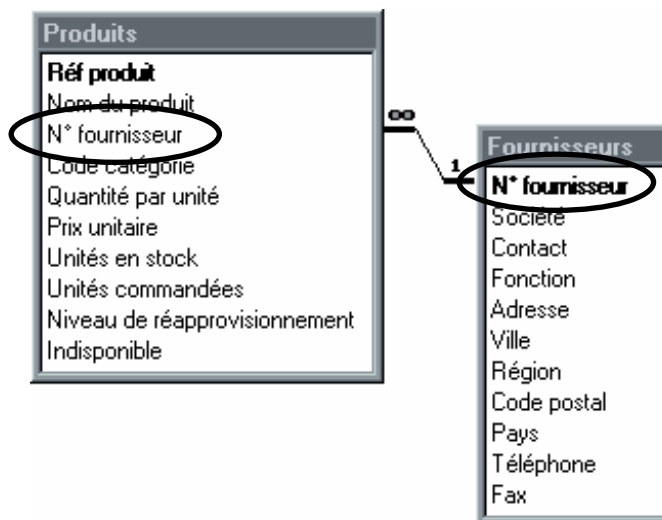
Liste de choix

Un nom de champ peut compter jusqu'à 64 caractères, espaces inclus. Pour obtenir de l'aide, appuyez sur F1.

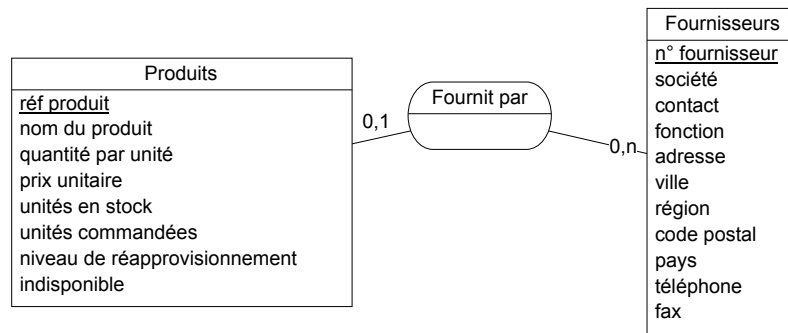
4.B.3 Jointures

Une jointure un est lien logique existant entre deux tables qui permet d'établir une correspondance entre les lignes des ces tables. Par exemple, le N° fournisseur dans la table Produits correspond au N° fournisseur de la table Fournisseurs. Autrement dit, à chaque produit correspond un fournisseur.

Cette jointure est représentée de la façon suivante dans Access :



Elle correspond, dans le schéma conceptuel (MCD), à l'association « Fournit par » suivante :



Access ne présente pas les cardinalités comme dans un MCD. On retrouve uniquement les valeurs maximum (1 et n) qui sont représentées respectivement par les signes 1 et ∞ (infini). On note au passage que la position des signes est inversée, la signification reste toutefois identique.

On ne retrouve pas les valeurs minimum (0 et 1). En fait, ces valeurs minimum sont mémorisées sous la forme d'une propriété dans la base de données. Si la cardinalité minimum est 0, la colonne qui permet la jointure peut ne pas contenir de valeur, autrement dit, il peut ne pas exister de correspondance avec une ligne d'une autre table. Alors que si la cardinalité minimum est 1, la colonne doit toujours prendre une valeur.

Dans les SGBDR, on parle de valeur « Null » pour désigner le fait qu'une donnée n'a pas de valeur. Null n'est pas la valeur 0 ni une chaîne de caractères vide, on peut considérer qu'il s'agit d'un symbole de « non valeur ».

Les champs d'une table Access possèdent une propriété nommée « Null interdit ». Si cette propriété vaut Vrai (valeur Oui), le champ doit toujours contenir une valeur, sinon il peut ne pas contenir de valeur. C'est justement le cas du N° fournisseur dans la table Produits : il peut ne pas prendre de valeur, donc la correspondant (la jointure) avec la table fournisseur peut ne pas exister et la cardinalité minimale de l'association vers cette table est bien 0.

?	Nom du champ	Type de données	Description
	Réf produit	NuméroAuto	Numéro automatiquement assigné aux nouveaux produits.
	Nom du produit	Texte	
	N° fournisseur	Numérique	Identique au N° fournisseur de la table Fournisseurs.
	Code catégorie	Numérique	Identique au Code catégorie de la table Catégories.
	Quantité par unité	Texte	(par exemple : bouteille 1 litre x 12).
	Prix unitaire	Monétaire	
	Unités en stock	Numérique	
	Unités commandées	Numérique	
	Niveau de réapprovisionnement	Numérique	Nombre minimum d'unités à garder en stock.
	Indisponible	Oui/Non	'Oui' signifie que le produit n'est plus disponible.

Propriétés du champ

Général

Taille du champ

Format

Décimales

Masque de saisie

Légende

Valeur par défaut

Valide si

Message si erreur

Null interdit

Indexé

Liste de choix

Entier long
Auto
N° fournisseur
GenUniqueID()
Non
Oui - Avec doublons

Un nom de champ peut compter jusqu'à 64 caractères, espaces inclus. Pour obtenir de l'aide, appuyez sur F1.

4.B.4 Tableau de correspondance MCD / schéma physique Access

Le tableau de correspondance suivant montre comment Access représente les associations d'un MCD dans son schéma physique.

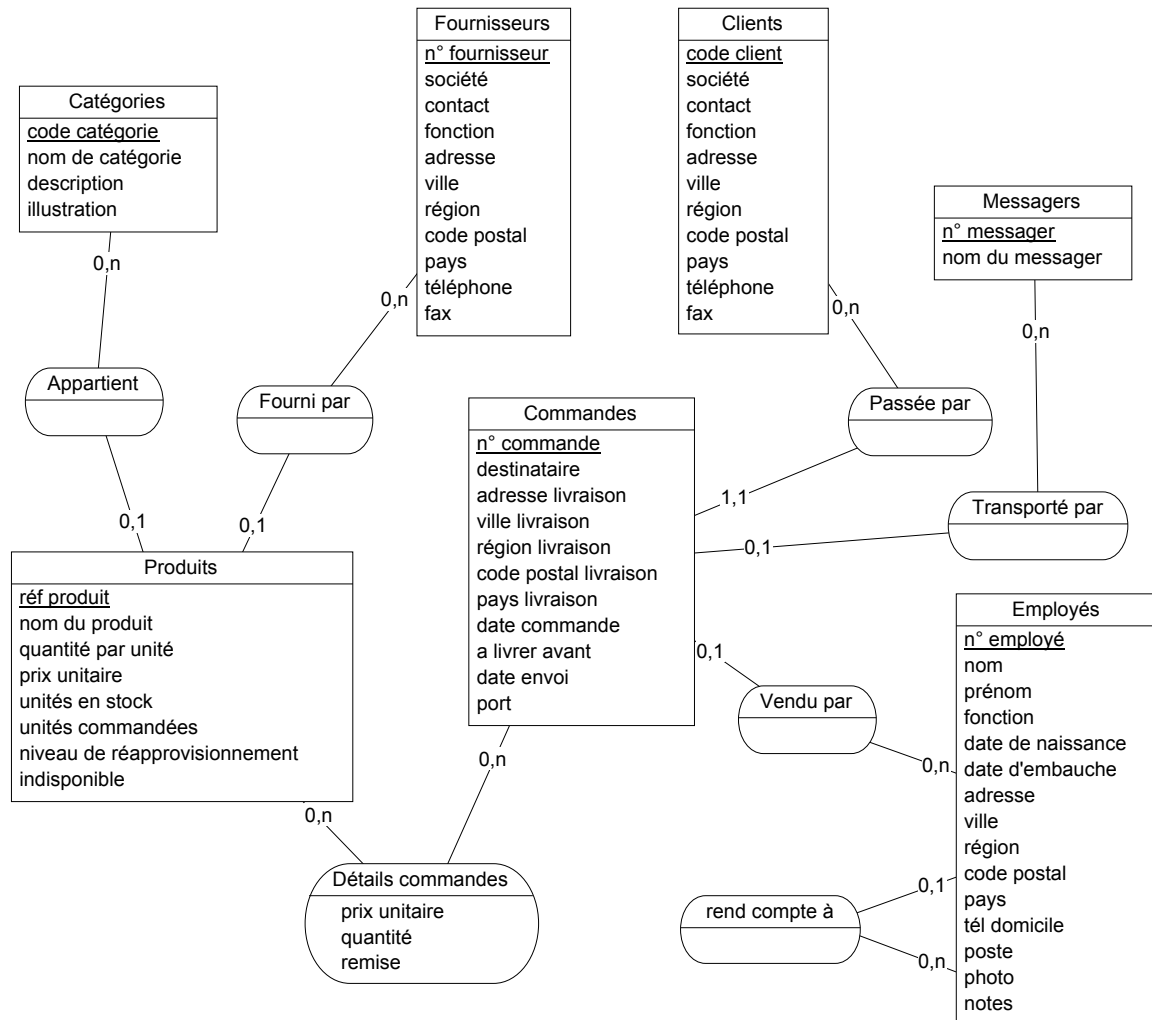
Type d'association (cardinalités maximum)	MCD	Schéma physique Access
1-1	<p>The MCD diagram shows two entity sets: 'Produits' and 'Produits2'. 'Produits' has attributes: <u>réf produit</u>, nom du produit, quantité par unité, prix unitaire, unités en stock, unités commandées, niveau de réapprovisionnement, indisponible. 'Produits2' has attributes: <u>réf produit2</u>, nom du produit, quantité par unité, prix unitaire, unités en stock, unités commandées, niveau de réapprovisionnement, indisponible. They are connected by a relationship 'correspond' with cardinalities 1,1 on both sides.</p>	<p>The Access physical schema shows two tables: 'Produits' and 'Produits2'. 'Produits' has fields: Réf produit (primary key), Nom du produit, N° fournisseur, Code catégorie, Quantité par unité, Prix unitaire, Unités en stock, Unités commandées, Niveau de réapprovisionnement, Indisponible. 'Produits2' has fields: Réf produit (primary key), Nom du produit, N° fournisseur, Code catégorie, Quantité par unité, Prix unitaire, Unités en stock, Unités commandées, Niveau de réapprovisionnement, Indisponible. A 1-1 relationship is shown between the 'Réf produit' fields of the two tables.</p>
1-N	<p>The MCD diagram shows two entity sets: 'Catégories' and 'Produits'. 'Catégories' has attributes: <u>code catégorie</u>, nom de catégorie, description, illustration. 'Produits' has attributes: <u>réf produit</u>, nom du produit, quantité par unité, prix unitaire, unités en stock, unités commandées, niveau de réapprovisionnement, indisponible. They are connected by a relationship 'Appartient' with cardinalities 0,n on the 'Catégories' side and 0,1 on the 'Produits' side.</p>	<p>The Access physical schema shows two tables: 'Catégories' and 'Produits'. 'Catégories' has fields: Code catégorie (primary key), Nom de catégorie, Description, Illustration. 'Produits' has fields: Réf produit (primary key), Nom du produit, N° fournisseur, Code catégorie, Quantité par unité, Prix unitaire, Unités en stock, Unités commandées, Niveau de réapprovisionnement, Indisponible. A 1-N relationship is shown between the 'Code catégorie' field of 'Catégories' and the 'Code catégorie' field of 'Produits'.</p>
N-N	<p>The MCD diagram shows two entity sets: 'Produits' and 'Commandes'. 'Produits' has attributes: <u>réf produit</u>, nom du produit, quantité par unité, prix unitaire, unités en stock, unités commandées, niveau de réapprovisionnement, indisponible. 'Commandes' has attributes: <u>n° commande</u>, destinataire, adresse livraison, ville livraison, région livraison, code postal livraison, pays livraison, date commande, à livrer avant. They are connected by a relationship 'Détails commandes' with cardinalities 0,n on both sides.</p>	<p>The Access physical schema shows three tables: 'Produits', 'Détails commandes', and 'Commandes'. 'Produits' has fields: Réf produit (primary key), Nom du produit, N° fournisseur, Code catégorie, Quantité par unité, Prix unitaire. 'Détails commandes' has fields: N° commande (primary key), Réf produit (foreign key), Prix unitaire, Quantité, Remise (%). 'Commandes' has fields: N° commande (primary key), Code client, N° employé. A 1-N relationship is shown between 'Produits' and 'Détails commandes' on the 'Réf produit' field. A 1-N relationship is shown between 'Détails commandes' and 'Commandes' on the 'N° commande' field.</p>

On note que les clés étrangères qui correspondent aux associations de type 1- N sont présentes dans les tables alors qu'elles n'apparaissent pas dans l'entité correspondante du MCD. Le nom des associations de type 1-N n'est plus utilisé dans le schéma physique. 1 et N représentent les cardinalités maximum de chaque côté de l'association.

Les associations de type N-N deviennent des tables dans la base de données. Ces tables ont pour clé primaire l'ensemble des clés primaires issues des tables reliées par l'association.

Dans le schéma physique d'une base Access, les liens partent et arrivent sur les attributs qui assurent la jointure entre les tables.

Voici le MCD complet qui correspond à la base Comptoir, remarquez que le champ « Rend compte à » de la table Employés devient une association réflexive. En effet, « rend compte à » est le numéro du chef de l'employé qui est lui-même un employé présent dans la table Employés.



4.C. Intégrité d'une base de données

L'une des missions essentielles des SGBD est d'assurer à tout instant l'intégrité c'est à dire la cohérence, la fiabilité, la pertinence des données qu'il gère.

Le concepteur de la base de données a la responsabilité de définir les contraintes d'intégrité qui s'appliquent à la base au moment de sa création. En cours d'utilisation le système vérifie en permanence le respect de ces contraintes.

4.C.1 Intégrité de domaine

Le domaine d'un attribut est l'ensemble des valeurs qu'il peut prendre. L'intégrité de domaine vérifie que les données de la base ne prennent que des valeurs autorisées.

Par exemple, dans la table Clients le code client doit avoir une longueur maximum de 5 caractères (taille du champ) qui doivent être 5 lettres

majuscules (masque de saisie). Ici, le domaine du champ est l'ensemble des chaînes de cinq lettres majuscule, Access ne permettra pas la saisie d'un autre type de valeur. Pour les champs numériques, le domaine peut être un intervalle de valeurs.

Nom du champ	Type de données	Description
Code client	Texte	Code unique de cinq caractères basé sur le nom du client.
Société	Texte	
Contact	Texte	

Propriétés du champ

Général
Liste de choix

Taille du champ	5
Format	
Masque de saisie	>LLLLL
Légende	Code client
Valeur par défaut	
Valide si	
Message si erreur	
Null interdit	Non
Chaîne vide autorisée	Non
Indexé	Oui - Sans doublons

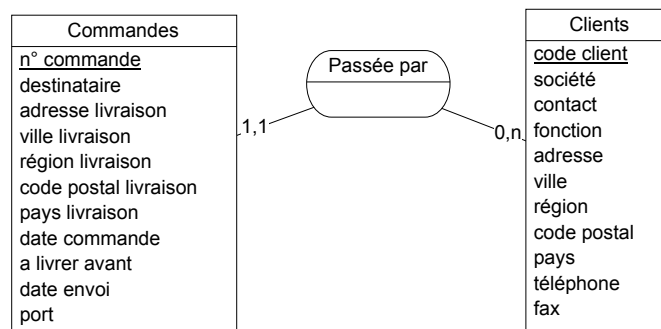
Un nom de champ peut compter jusqu'à 64 caractères, espaces inclus. Pour obtenir de l'aide, appuyez sur F1.

4.C.2 Intégrité référentielle

L'intégrité référentielle permet de contrôler la cohérence des données en vérifiant les contraintes de jointure entre les tables.

a. Mise en œuvre de l'intégrité référentielle

Prenons pour exemple l'association « Passée par » entre les tables Commandes et Clients. Cette association indique qu'à toute commande doit correspondre exactement un client.



Voyons comment cette contrainte d'intégrité référentielle est mise en œuvre dans Access.

Observons tout d'abord les valeurs contenues dans les tables et vérifions la correspondance :

N° commande	Code client	Employé	Date commande
10360	BLONP	Peacock, Margaret	19-nov-93
10361	QUICK	Davolio, Nancy	19-nov-93
10362	BONAP	Leverling, Janet	22-nov-93
10363	DRACD	Peacock, Margaret	23-nov-93
10364	EASTC	Davolio, Nancy	23-nov-93
10365	ANTON	Leverling, Janet	24-nov-93
10366	GALED	Callahan, Laura	25-nov-93

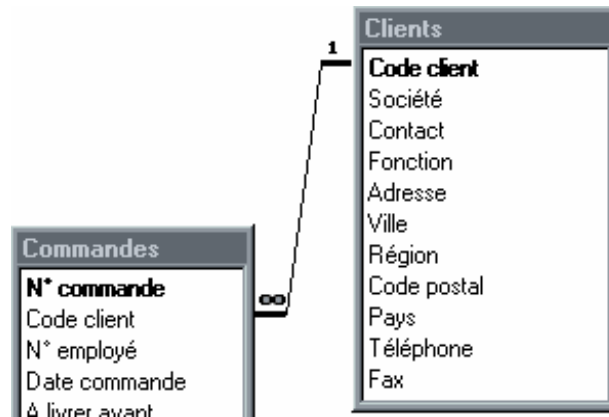
Enr: 1 sur 830

Code client	Société	Contact
ANTON	Antonio Moreno Taquería	Antonio Moreno
AROUT	Around the Horn	Thomas Hardy
BERGS	Berglunds snabbköp	Christina Berglund
BLAUS	Blauer See Delikatessen	Hanna Moos
BLONP	Blondel père et fils	Frédérique Citeaux
BOLID	Bólido Comidas preparadas	Martín Sommer
BONAP	Bon app'	Laurence Lebihan

Enr: 1 sur 91

Pour assurer le respect de cette correspondance, Access permet de créer un lien entre les deux tables en allant de la clé étrangère vers la clé primaire.

*Cliquer sur Code client
dans Commandes,
déplacer jusqu'à Code
client dans Clients,
lâcher.*



Ensuite, dans les propriétés de ce lien, la vérification de l'intégrité référentielle est demandée en cochant la case correspondante.

*Double clic sur le lien
dans la fenêtre
Relations*

Table/Requête:	Table/Requête liée:
Clients	Commandes
Code client	Code client

☒ Appliquer l'intégrité référentielle

☒ Mettre à jour en cascade les champs correspondants

☐ Effacer en cascade les enregistrements correspondants

Type de relation: Un à plusieurs

Enfin, pour assurer une cardinalité minimum à 1, on valide la propriété « Null interdit » pour le champ Code client qui est clé étrangère dans la table Commandes.

BDD / Tables /
Commandes /
Modifier / Cliquer sur
Code client

Nom du champ	Type de données	Description
N° commande	NuméroAuto	Numéro d'ordre unique des commandes.
Code client	Texte	Identique au Code client de la table Clients.
N° employé	Numérique	Identique au N° employé de la table Employés.

Propriétés du champ

Général

Taille du champ: 5

Format:

Masque de saisie:

Légende: Code client

Valeur par défaut:

Valide si:

Message si erreur:

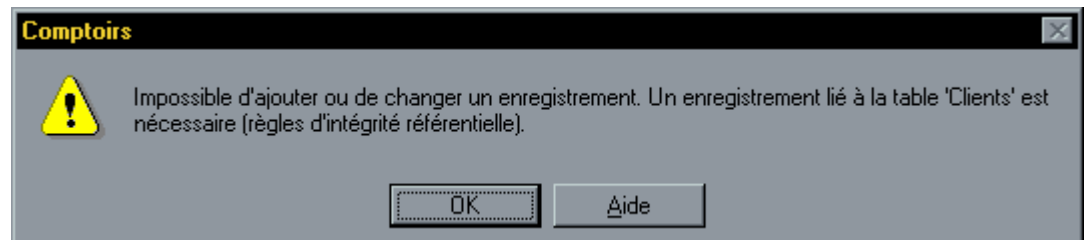
Null interdit: Oui

Chaîne vide autorisée: Non

Indexé: Oui - Avec doublons

Si Oui, une valeur sera requise pour ce champ.

Vérifions le respect de l'intégrité référentielle. Quand nous essayons de saisir une nouvelle commande ou de modifier une commande avec un code client inexistant dans la table Clients, Access affiche le message suivant :



b. Options de l'intégrités référentielles

Deux options particulières permettent de compléter le paramétrage de l'intégrité référentielle.

Relations

Table/Requête:	Table/Requête liée:	
Clients	Commandes	
Code client	Code client	

OK

Annuler

Type jointure...

☒ Appliquer l'intégrité référentielle

☒ Mettre à jour en cascade les champs correspondants
☐ Effacer en cascade les enregistrements correspondants

Type de relation: Un à plusieurs

- L'option « Mettre à jour en cascade les champs correspondants » permet, quand on modifie la valeur d'une clé primaire, de modifier automatiquement les valeurs correspondantes des clés étrangères de la table liée. Par exemple, si l'on modifie un code client dans la table Clients, toutes les références à ce client dans la table Commandes seront également modifiées.
- L'option « Effacer en cascade les enregistrements correspondants » consiste à demander la suppression automatique des enregistrement liés dès lors qu'un enregistrement est supprimé. Par exemple, si cette option est validée, la suppression d'un client dans la table clients supprimera automatiquement toutes les commandes de ce client dans la table Commandes.

Ces deux options doivent être manipulées avec prudence. Les modifications qu'elles entraînent peuvent se répercuter sur plusieurs tables.

c. Clés étrangères et liste de valeurs

Une clé étrangère est un attribut d'une table qui fait référence à une ligne d'une autre table. Par exemple, puisqu'à chaque produit correspond un fournisseur, il existe un champ N° fournisseur dans la table Produits. Ce champ est clé étrangère car il correspond à la clé primaire de la table Fournisseur.

Dans la mesure où l'intégrité référentielle implique que seul un numéro de fournisseur existant dans la table Fournisseurs puisse être saisi dans la table Produits, il semble judicieux de proposer à l'utilisateur une liste des valeurs existantes de N° fournisseur tirées de la table Fournisseurs.

Dans l'illustration ci-après, on voit comment cette saisie s'opère avec Access :

On veut modifier le fournisseur du produit, Access propose une liste de valeurs possibles

Réf produit	Nom du produit	N° fournisseur	Code
1	Chai	Exotic Liquids	
2	Chang	Exotic Liquids	
3	Aniseed Syrup	Exotic Liquids	
4	Chef Anton's Cajun Seasoning	Exotic Liquids	
5	Chef Anton's Gumbo Mix	New Orleans Cajun Delights	
6	Grandma's Boysenberry Spread	Grandma Kelly's Homestead	
7	Uncle Bob's Organic Dried Pears	Tokyo Traders	
8	Northwoods Cranberry Sauce	Cooperativa de Quesos 'Las Cabras	
9	Mishi Kobe Niku	Mayumi's	

Enr: 3 sur 77

Identique au N° fournisseur de la table Fournisseurs.

Exotic Liquids
Exotic Liquids
Exotic Liquids
Exotic Liquids
New Orleans Cajun Delights
Grandma Kelly's Homestead
Tokyo Traders
Cooperativa de Quesos 'Las Cabras
Mayumi's
Pavlova, Ltd.
Specialty Biscuits, Ltd.

La liste de valeurs est tirée d'une requête qui donne l'ensemble des noms de fournisseurs existants dans la table Fournisseurs

Nom du champ	Type de données	Description
Réf produit	NuméroAuto	Numéro automatique
Nom du produit	Texte	
N° fournisseur	Numérique	Identique au N° fournisseur
Code catégorie	Numérique	Identique au Code

Propriétés du champ

Général

Liste de choix

Afficher le contrôle: Zone de liste

Origine source: Table/requête

Contenu: Select [N°fournisseur], société from fournisseurs

Colonne liée: 1

Nbre colonnes: 2

En-têtes colonnes: Non

Largeurs colonnes: 0cm;4cm

Ces paramètres permettent d'afficher le nom du fournisseur qui est plus parlant qu'un simple numéro. C'est bien le n° de fournisseur qui est stocké dans la table (colonne liée).

4.D. Sécurité d'une base de données

Sécuriser une base de données consiste à garantir la pérennité et la confidentialité des données qu'elle contient.

4.D.1 Sécurité de l'accès aux données

A l'instar d'un système d'exploitation, un SGBD se doit d'identifier des utilisateurs de façon à leurs accorder des permissions d'accès au données.

Une première solution rudimentaire consiste à protéger toute la base de données par un mot de passe. Ce mot de passe sera demandé à tout utilisateur dès qu'il ouvre la base. Cette méthode est insuffisante car s'il existe plusieurs utilisateurs, ceux-ci doivent connaître le même mot de passe d'où un risque important de diffusion. Par ailleurs, le mot de passe donne un accès complet à la base ce qui n'est pas forcément nécessaire pour tous les utilisateurs.

Outils / Sécurité /
Définir le mot de
passe de la base de
données

The screenshot shows a dialog box titled "Définir le mot de passe de la base de données". It contains two text input fields: "Mot de passe:" and "Confirmation:". To the right of these fields are two buttons: "OK" and "Annuler".

a. Utilisateurs et groupes d'utilisateurs

Access, comme tous les SGBD, permet de définir des utilisateurs en leur attribuant un nom et un mot de passe. Dès que des utilisateurs existent, il est possible de leur attribuer des autorisations précises pour accéder aux données.

Outils / Sécurité /
Gestion des
utilisateurs et des
groupes

The screenshot shows a complex dialog box titled "Gestion des utilisateurs et des groupes". It has three tabs at the top: "Utilisateurs", "Groupes", and "Changer le mot de passe". The "Utilisateurs" tab is selected. Inside, there is a section for "Utilisateur" with a "Nom:" field containing "Eric" and a dropdown arrow. Below this are three buttons: "Nouveau...", "Supprimer", and "Effacer le mot de passe". Another section is labeled "Membre du groupe". It contains two list boxes: "Liste des groupes:" (with "Administrateurs" and "Utilisateurs" listed, "Administrateurs" is selected) and "Utilisateur inscrit dans:" (with "Utilisateurs" listed). Between these lists are two buttons: "Inscrire >>" and "<< Enlever". At the bottom of the dialog is a button labeled "Imprimer les utilisateurs et les groupes".

Afin de faciliter l'octroi de ces autorisations, Access autorise la création de groupe d'utilisateurs. Ces groupes disposent également de droits. Quand un utilisateur appartient à un groupe, il hérite des droits accordés à ce groupe.

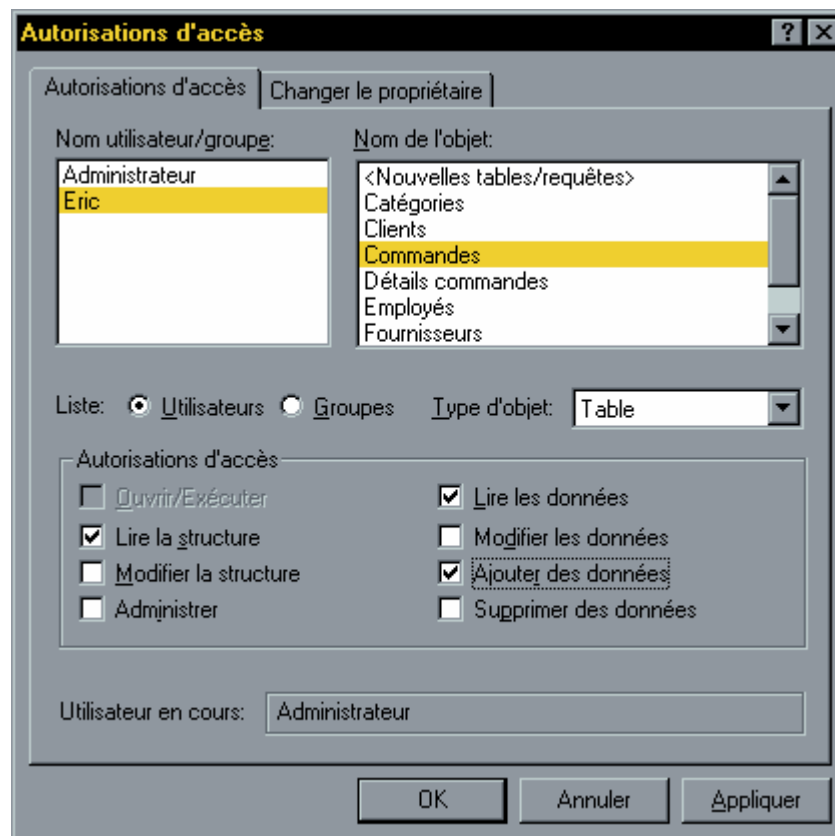
Dans la copie d'écran ci-dessus, l'utilisateur « Eric » fait partie du groupe « Utilisateurs ». Il existe actuellement deux groupes : « Utilisateurs » et « Administrateurs ».

Par défaut, après installation, Access ne connaît qu'un utilisateur : l'administrateur nommé « Administrateur » et celui-ci n'a pas de mot de passe. Dans ce cas, Access ne demande aucun mot de passe, mais dès que l'administrateur possède un mot de passe, celui-ci est demandé dès le démarrage du logiciel.

b. Autorisation d'accès aux données

Les autorisations possibles sont fonction du type d'objet sur lequel les droits sont accordés. Dans la copie d'écran ci-après on trouve les autorisations qui peuvent être accordées à une table (ici la table Commandes).

Outils / Sécurité /
Autorisations d'accès



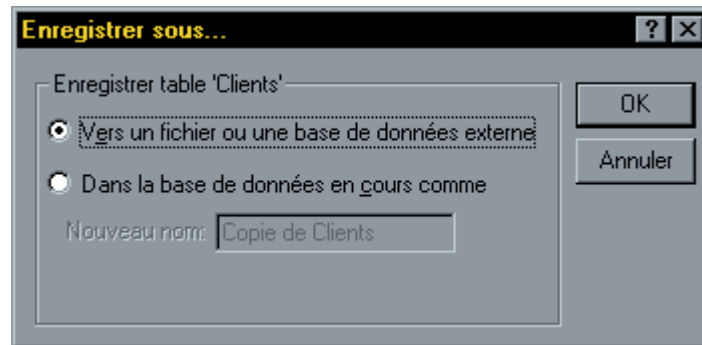
4.D.2 Sécurité des données

a. Sauvegarde, exportation

L'opération de sauvegarde consiste à dupliquer tout ou partie d'une base de données de façon à disposer d'un deuxième exemplaire des données en cas d'altération ou de perte de l'original.

La sauvegarde peut aussi permettre l'exportation des données dans un format lisible par d'autres logiciels.

BDD / Tables /
Clients / Menu Fichier
/Enregistrer sous,
Exporter



b. Réplication

La réplication permet d'effectuer des copies particulières d'une base de données appelées « réplicas » de façon à ce que des utilisateurs situés à divers endroits puissent tous travailler à partir de leur propre copie au même moment. La réplication est différente de l'exécution de simples copies de fichiers de base de données parce que des réplicas peuvent être synchronisés entre eux. Lorsqu'une base de données est répliquée, la base de données d'origine devient le réplica-maître du jeu de réplicas.

4.E. Requêtes

Une requête est un ordre adressé à un SGBD. Cet ordre peut consister à extraire, à ajouter, à modifier, à administrer les données de la base. De façon générale, l'utilisateur, comme l'administrateur, dialogue avec le SGBD en lui soumettant des requêtes (des questions) et en récupérant en retour des résultats (les réponses).

Voyons un exemple de requête à soumettre à Access sur la base de données « Les comptoirs ». La requête est libellé ainsi en français : « liste des numéros de commandes et nom de société qui ont passé la commande ».

Voici le résultat :

N° commande	Société
10692	Alfreds Futterkiste
10702	Alfreds Futterkiste
10835	Alfreds Futterkiste
10952	Alfreds Futterkiste
11011	Alfreds Futterkiste
10308	Ana Trujillo Emparedados y helados
10625	Ana Trujillo Emparedados y helados
10759	Ana Trujillo Emparedados y helados
10926	Ana Trujillo Emparedados y helados
10365	Antonio Moreno Taquería

Enr. 1 sur 830

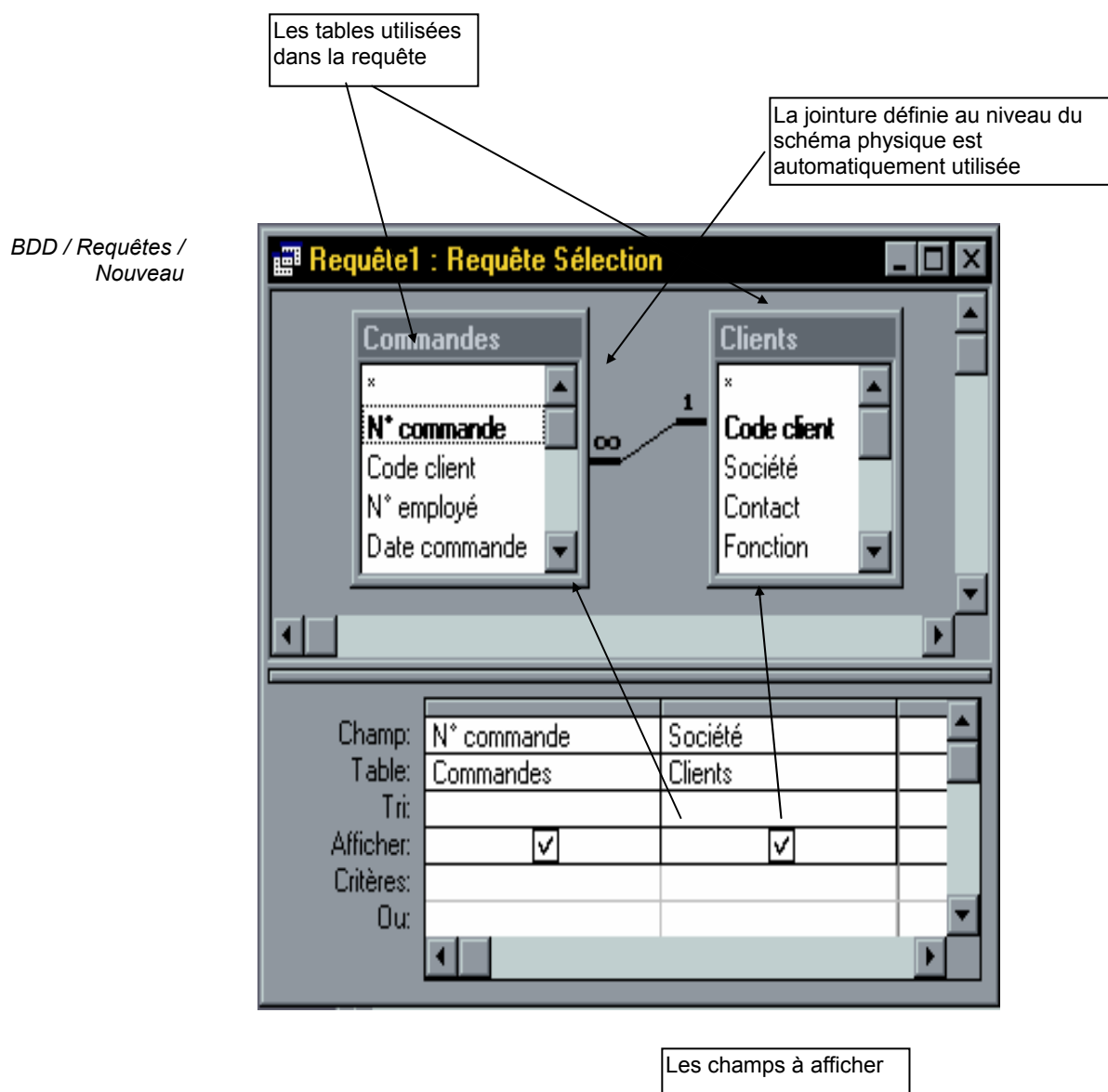
Ces données sont tirées des tables Commandes et Clients en utilisant la jointure sur le code client qui a été montrée plus haut.

Il existe différentes façon de soumettre des requêtes à un SGBDR.

4.E.1 En mode graphique

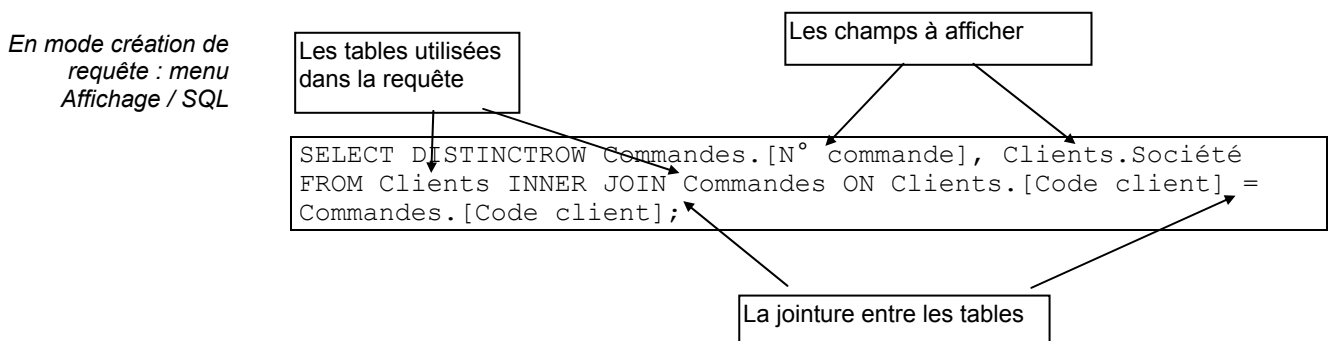
Dans les environnements graphiques, le mode graphique est naturellement privilégié. Il s'agit pour l'essentiel de sélectionner à l'aide de la souris les éléments sur lesquels on souhaite travailler. On parle aussi de « requête par l'exemple » (*Query by example* - QBE).

Voici comment la requête ci dessus a été construite en mode graphique avec Access :



4.E.2 En langage SQL

Le langage SQL est spécialement conçu pour cet usage. Écrire une requête SQL consiste à rédiger du texte ce qui nécessite de bien connaître le langage ainsi que l'organisation (la structure) de la base de données.



4.E.3 La programmation

La plupart des SGBDR permettent l'usage d'un langage de programmation procédural pour répondre à des besoins de développement plus pointus. Cette dernière solution consiste souvent à insérer des requêtes SQL au sein de programmes écrits en Basic, C ou COBOL.

Voici la même requête exprimée en langage Visual Basic dans un module de la base « Les comptoirs » :

BDD / Module /
Nouveau / Saisir le texte
ci-contre.

Pour exécuter :
Affichage / fenêtre de
debugage

Entrer
« ListeDesCommandes
» puis appuyer sur la
touche Entrée

```
Sub ListeDesCommandes()
    'Déclare un objet base de donnée
    Dim Base As DATABASE
    'Déclare un objet résultat de requête (jeu d'enregistrements)
    Dim Résultat As Recordset
    'déclare une chaîne de caractère qui contient la requête SQL
    Dim Requête As String

    'Ouvre la base de données
    Set Base = CurrentDb()
    'Affecte la requête SQL
    Requête = "SELECT DISTINCTROW Commandes.[N° commande],
Clients.Société FROM Clients INNER JOIN Commandes ON Clients.[Code
client] = Commandes.[Code client];"
    'Soumet la requête à la base et récupère les lignes du résultat
    'dans l'objet Résultat
    Set Résultat = Base.OpenRecordset(Requête)
    'Tant qu'il y a des lignes dans l'objet Résultat...
```

```
'Affiche les données dans la fenêtre de débogage
Do While Not Résultat.EOF
    Debug.Print Résultat![N° commande], Résultat!Société
    Résultat.MoveNext
Loop
End Sub
```

4.F. Les outils de développement

Les utilisateurs n'ont pas à connaître la structure de la base de données ni toutes les contraintes d'intégrité mises en place. Ils n'ont pas à connaître l'existence de tables, ni même de requêtes. Ils souhaitent simplement sélectionner une commande dans un menu ou bien cliquer sur un bouton pour obtenir les liste des sociétés par commande sur leur écran ou sur papier.

Pour le développeur d'application, ceci signifie qu'il doit construire une interface entre la base de données et l'utilisateur. Les éditeurs de SGBD proposent tous des outils de développement adaptés à leurs produits.

Ces outils permettent notamment la création de dialogues sur écran et d'états imprimés.

4.F.1 Les formulaires sur écran

Avec Access, les dialogues sur écran sont bâtis à l'aide de formulaires. Chaque formulaire est une fenêtre au sens Windows qui gère le dialogue entre l'utilisateur et l'application.

La formulaire suivant permet de gérer les employés de la base « Comptoir ».

BDD / Formulaires /
Employés / Ouvrir

Numéro de l'employé, il n'est pas modifiable, c'est la clé primaire de la table !

Nom et prénom de l'employé

Nancy Davolio

N° employé: 1

Prénom: Nancy

Nom: Davolio

Fonction: Représentant(e)

Rend compte à: Fuller, Andrew

Date d'embauche: 01-mai-92

Extension: 5467

Boutons pour voir la suite des informations sur l'employé

Informations personnelles

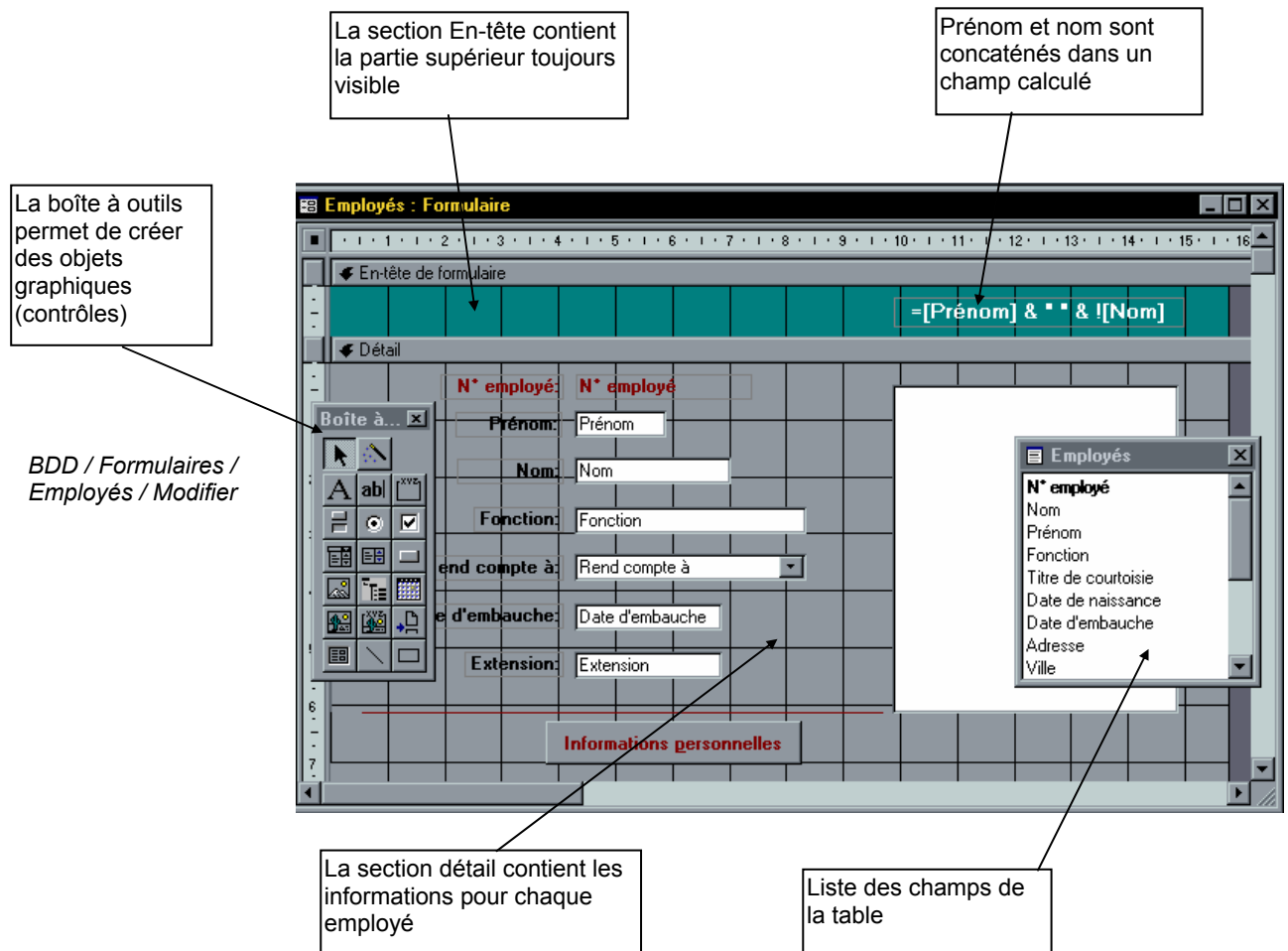
Ent: 1 sur 9

Boutons pour passer d'un employé à l'autre

Ce bouton permet d'ajouter un nouvel employé

Cette zone de liste permet de choisir une valeur car la donnée est clé étrangère

Voici maintenant le même formulaire en phase de développement :




4.F.2 Les états imprimés

Les états permettent de produire des documents imprimés qui présente l'information sur papier.

Voici, par exemple, une facture :

BDD / Etats /
Facture / Aperçu



LES COMPTOIRS

FACTURE

Parc d'activités La Grenouillère 15 rue du Moulin, 44000 Nantes Tél (33) 08.80.22.22 Fax (33) 08.80.20.20

Envoyé à: Save-a-lot Markets
187 Suffolk Ln.
Boise ID 83720
Etats-Unis

Facturé à: Save-a-lot Markets
187 Suffolk Ln.
Boise ID 83720
Etats-Unis

N° com.:	Code client:	Représentant:	Date com.:	Livrer avant:	Date d'env.:	Message:
11064	SAVEA	Nancy Davolio	28-avr-95	26-mai-95	01-mai-95	Speedy Express

Réf produit:	Nom du produit:	Qté:	Prix unitaire:	Remise (%):	Total:
17	Alice Mutton	77	195,00 F	10 %	13 513,50 F
41	Jack's New England Clam Chowder	12	48,25 F	0 %	579,00 F
53	Perth Pasties	25	164,00 F	10 %	3 690,00 F
55	Pâté chinois	4	120,00 F	10 %	432,00 F
68	Scottish Longbreads	55	62,50 F	0 %	3 437,50 F

Sous-total:	21 652,00 F
Port:	150,45 F
Total:	21 802,45 F


Voici le même état en phase de développement :

BDD / Etats / Facture
/ Modifier

La section en-tête de page contient le titre de l'état

La section entête de commande est crée pour chaque numéro de commande

La date est obtenue à l'aide d'un champ calculé

En-tête de page															
										FACTURE					
Parc d'activités La Grenouillère 15, rue du Moulin, 44000 Nantes Tél (33) 08.80.22.22 Fax (33) 08.80.20.20										Date: [Date, réduit"]					
En-tête de groupe N° commande															
Envoyé à : Destinataire Adresse livraison =VraiFaux([Région] Est Null;[Ville] livrai Pays livraison								Facturé à : Clients.Société Adresse =VraiFaux([Région] Est Null;[Ville] & "" Pays							
N° com.:	Code client:	Représentant:	Date com.:	Livrer avant:	Date d'envoi:	Message:									
comman	Code client	PersonneVentes	ate commande	A livrer avant	Date envoi	Jom du message									
Réf produit:	Nom du produ	Qté:	Prix unitaire	Remise (%)	Total:										
Détail															
Réf proc	Nom du produit	Quantité	Prix unitaire	Remise (%)	PrixEtendu										
Pied de groupe N° commande															
										Sous-total: [me([PrixEtendu])]					
										Port: Port					
										Total: [ta[Facture]+[Port]]					

La section détail définit l'affichage de chaque ligne de la facture

Des champs calculés sont utilisés pour obtenir les totaux

5. LE LANGAGE SQL

SQL (*Structured Query Language*) a une triple fonction :

- Interrogation et modification d'une base de données relationnelle
- Définition et modification du schéma d'une base de données relationnelle
- Contrôle de sécurité et d'intégrité de la base.

5.A. Le standard SQL

En principe, le standard Ansi est reconnu seulement aux Etats-Unis, l'équivalent français est la norme Afnor. La norme internationale de SQL est la norme ISO (International Standards Organisation) numéro 9075 qui date de 1987. Les normes sont accompagnées de niveaux qui indiquent le degré d'évolution de SQL.

L'ISO a défini les normes et les niveaux suivants :

SQL89

SQL92

Entry level

Intermediate level

Full level

SQL3

La norme définit deux langages SQL :

- un langage de définition de données, **SQL-LDD**, à utiliser pour déclarer les structures logiques de données et leurs contraintes d'intégrité ;
- un langage de manipulation de données et de modules, **SQL-LMD**, pour déclarer les procédures d'exploitation et les appels à utiliser dans les programmes. On peut en plus, distinguer la partie interrogation (LID) de la partie mise à jour.

Cependant, on ajoute souvent une troisième composante pour la gestion des accès aux données : le langage de contrôle de données : **SQL-LCD**

Les principales instructions du langage SQL			
LMD		LDD	
LID		LCD	
SELECT	INSERT	GRANT	CREATE
	UPDATE	REVOKE	ALTER
	DELETE		TRUNCATE
			DROP
			RENAME

Oracle version 7.1 a été certifié par le National Institute of Standards and Technology, à 100% de conformité avec le standard SQL92-Entry Level. Oracle 7 est aussi totalement conforme au standard du gouvernement américain FIPS 127-2.

Oracle inclut une fonctionnalité signalant toute utilisation de fonctions non conformes au standard SQL.

5.B. L'ALGÈBRE RELATIONNELLE

5.B.1 Introduction

L'algèbre relationnelle est un outil théorique défini par CODD dans le cadre de son modèle relationnel. Cette algèbre permet de construire des requêtes sur des relations. Le principe fondamental consiste à considérer que l'information que l'on veut sélectionner par une requête peut s'exprimer sous la forme d'une relation. Cette relation est obtenue en appliquant successivement des opérateurs sur des opérandes qui sont des relations.

Ces opérandes sont soit le résultat d'autres opérations (relations provisoires), soit des relations de la base de données (relations permanentes).

Dans l'algèbre relationnelle, on ne manipule que des relations, à la fois comme données et comme résultats.

Le langage SQL est un outil pratique d'interrogation sur SGBDR qui reprend directement ou indirectement tous les opérateurs relationnels.

5.B.2 Les opérateurs algébriques

Pour allons illustrer les différents opérateurs algébriques à partir de la base COMPTOIR fournie avec Ms/Access

a. Projection

Définition : L'opération de projection permet de retenir uniquement certaines colonnes d'une table.

Exemple : soit la table Clients avec la définition en extension suivante :

N°	Nom	Prénom	Fonction	Titre
1	Davolio	Nancy	Représentant(e)	Mlle
2	Fuller	Andrew	Vice-Président	Dr.
3	Leverling	Janet	Représentant(e)	Mlle
4	Peacock	Margare	Représentant(e)	Mme
5	Buchanan	Steven	Chef des ventes	M.
6	Suyama	Michael	Représentant(e)	M.
7	King	Robert	Représentant(e)	M.
8	Callahan	Laura	Assistante commerciale	Mlle
9	Dodsworth	Anne	Représentant(e)	Mlle

Clients [Codeclient , Nomclient] correspond à la table ci-dessous :

N°	Nom
1	Davolio
2	Fuller
3	Leverling
4	Peacock
5	Buchanan
6	Suyama
7	King
8	Callahan
9	Dodsworth

Syntaxe algébrique	Représentation graphique
<p>$R = \text{Projection} (<\text{nom relation}> ; <\text{nom des attributs}>)$</p> <p>Exemple</p> <p>$R = \text{Projection} (\text{Clients}; \text{Codeclient}, \text{Nomclient})$</p>	

b. Sélection ou restriction

Définition : L'opération de sélection permet d'extraire d'une table les lignes qui satisfont une ou plusieurs conditions.

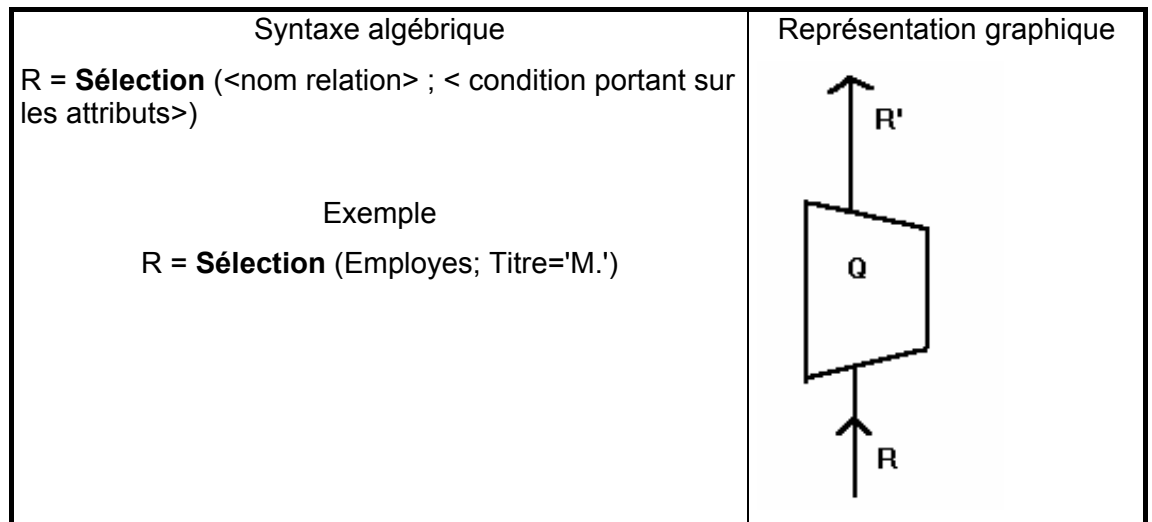
La condition est construite à partir des opérateurs logiques et des opérateurs de comparaison.

Exemple : soit la table Employés avec la définition en extension suivante :

N°	Nom	Prénom	Fonction	Titre
1	Davolio	Nancy	Représentant(e)	Mlle
2	Fuller	Andrew	Vice-Président	Dr.
3	Leverling	Janet	Représentant(e)	Mlle
4	Peacock	Margare	Représentant(e)	Mme
5	Buchanan	Steven	Chef des ventes	M.
6	Suyama	Michael	Représentant(e)	M.
7	King	Robert	Représentant(e)	M.
8	Callahan	Laura	Assistante commerciale	Mlle
9	Dodsworth	Anne	Représentant(e)	Mlle

On veut exprimer la requête suivante : afficher les employés dont le titre de courtoisie est égal à M. On obtient pour le résultat suivant :

N°	Nom	Prénom	Fonction	Titre
5	Buchanan	Steven	Chef des ventes	M.
6	Suyama	Michael	Représentant(e)	M.
7	King	Robert	Représentant(e)	M.



c. Jointure

Contrairement aux opérateurs précédents qui sont unaires, l'opération de Jointure agit sur 2 tables (ou relations) et peut se généraliser à n tables (ou relations).

Définition : L'opération de jointure permet de rapprocher des informations issues de deux tables distinctes qui ont un attribut commun (dont les valeurs sont prises dans le même domaine).

La correspondance s'établit le plus souvent entre une clé étrangère et une clé primaire au sens du modèle relationnel.

Exemple : Soient les deux tables Employés et Commandes avec les définitions en extension suivante

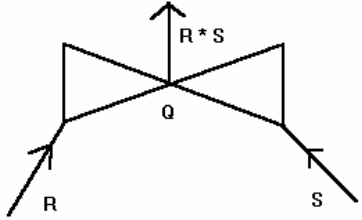
N°	Nom	Prénom	Titre
1	Davolio	Nancy	Mlle
2	Fuller	Andrew	Dr.
3	Leverling	Janet	Mlle
4	Peacock	Margare	Mme
5	Buchanan	Steven	M.
6	Suyama	Michael	M.
7	King	Robert	M.
8	Callahan	Laura	Mlle
9	Dodsworth	Anne	Mlle

N°	Code client	Employé	Date commande
10248	VINET	5	01-juil-93
10249	TOMSP	6	02-juil-93
10250	HANAR	4	05-juil-93
10251	VICTE	3	05-juil-93
10252	SUPRD	4	06-juil-93
10253	HANAR	3	07-juil-93

La jointure entre Commandes et Employés aura la définition en extension suivante :

N°	Code client	Employé	Date commande	Nom	Prénom	Titre
10248	VINET	5	01-juil-93	Buchanan	Steven	M.
10249	TOMSP	6	02-juil-93	Suyama	Michael	M.
10250	HANAR	4	05-juil-93	Peacock	Margaret	Mme
10251	VICTE	3	05-juil-93	Leverling	Janet	Mlle
10252	SUPRD	4	06-juil-93	Peacock	Margaret	Mme
10253	HANAR	3	07-juil-93	Leverling	Janet	Mlle

Interprétation : l'opérateur de Jointure permet entre autres de **réunir les propriétés** d'une entité unique qui avaient été **séparées par la normalisation**.

Syntaxe algébrique	Représentation graphique
<p>$R = \text{Jointure} (<\text{nom relation gauche}>, <\text{nom relation droite}>; <\text{condition de jointure portant sur les attributs}>)$</p> <p style="text-align: center;">Exemple</p> <p>$R = \text{Jointure} (\text{Commandes}, \text{Employes}; \text{Commandes.numemploye} = \text{Employes.numemploye})$</p>	

Exemple : jointure entre commandes et clients

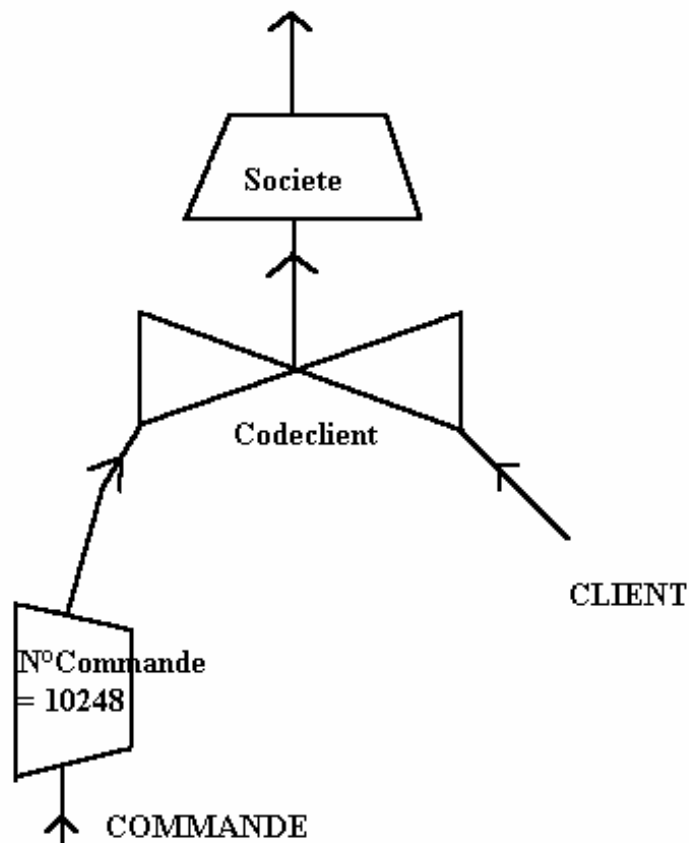
Si on veut formuler la requête :

"Quel est le nom du client (colonne société) ayant passé la commande numéro 10248 ?"

On ne peut se contenter d'appliquer des opérateurs sur la relation Commandes.

Il faut appliquer l'opérateur de Jointure sur les relations Commande et Clients en utilisant le constituant commun CodeClient.

Représentation graphique



d. Union

Définition : L'union de deux tables R et S, notée **R U S** ou **UNION(R,S)**, est une table T de même schéma, contenant les **tuples appartenant à R ou à S ou aux deux tables**.

Rappel : deux tables R et S ont le même schéma lorsque les **domaines** associés aux colonnes de R sont **identiques** aux domaines des colonnes de la table S.

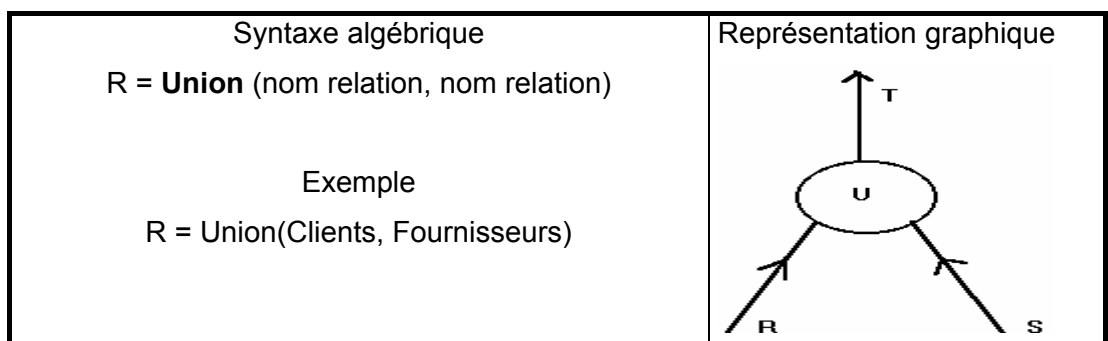
Exemple : soient les deux tables Clients (Codeclient, nomclient, adresse) et Fournisseurs (Codefour, nomfour, adresse) avec les contenus suivants :

Code client	Société	Adresse
ALFKI	Alfreds Futterkiste	Obere Str. 57
ANATR	Ana Trujillo Emparedados y helados	Avda. De la Constitución 2222
BERGS	Berglunds snabbköp	Berguvsvägen 8
BOTTM	Bottom-Dollar Markets	23 Tsawassen Blvd.
BSBEV	B's Beverages	Fauntleroy Circus

N° fournisseur	Société	Adresse
1	Exotic Liquids	49 Gilbert St.
2	New Orleans Cajun Delights	P.O. Box 78934
3	Grandma Kelly's Homestead	707 Oxford Rd.
4	Tokyo Traders	9-8 Sekimai
5	Cooperativa de Quesos 'Las Cabras'	Calle del Rosal 4
6	Mayumi's	92 Setsuko

L'union de Clients et Fournisseurs aura la définition en extension suivante :

Code client	Société	Adresse
ALFKI	Alfreds Futterkiste	Obere Str. 57
ANATR	Ana Trujillo Emparedados y helados	Avda. De la Constitución 2222
BERGS	Berglunds snabbköp	Berguvsvägen 8
BOTTM	Bottom-Dollar Markets	23 Tsawassen Blvd.
BSBEV	B's Beverages	Fauntleroy Circus
1	Exotic Liquids	49 Gilbert St.
2	New Orleans Cajun Delights	P.O. Box 78934
3	Grandma Kelly's Homestead	707 Oxford Rd.
4	Tokyo Traders	9-8 Sekimai
5	Cooperativa de Quesos 'Las Cabras'	Calle del Rosal 4
6	Mayumi's	92 Setsuko



e. Intersection

Définition : L'intersection de deux tables R et S, notée $R \cap S$ ou **INTERSECT(R,S)**, est une table T de même schéma, contenant les **tuples appartenant à la fois à R et S**

Rappel : deux tables R et S ont le même schéma lorsque les **domaines** associés aux colonnes de R sont **identiques** aux domaines des colonnes de la table S.

Exemple : Soient les deux tables qui correspondent à la liste des commandes (Numéro et date) passées avant le 27-aout-93 et comportant respectivement des produits de catégorie 2 et de catégorie 3 :

R = Commandes de catégorie 2

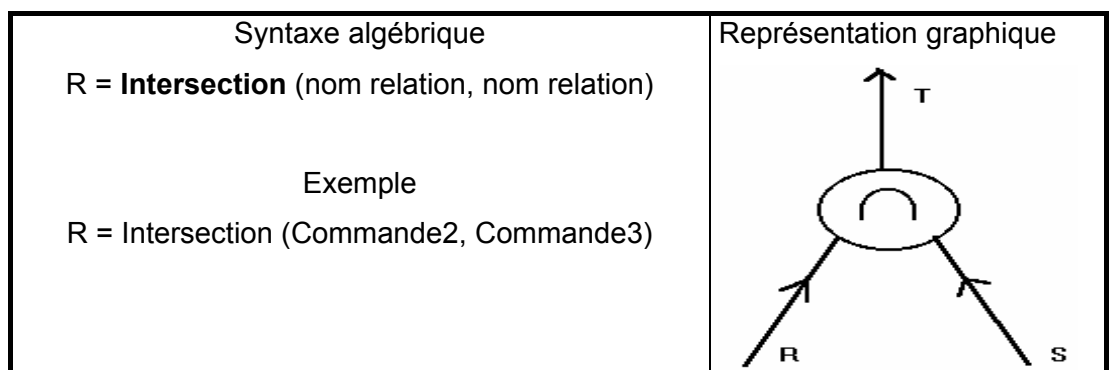
S = Commandes de catégorie 3

N° commande	Date commande
10250	05-juil-93
10251	05-juil-93
10256	12-juil-93
10257	13-juil-93
10258	14-juil-93
10262	19-juil-93
10278	09-août-93
10283	13-août-93
10284	16-août-93
10289	23-août-93
10290	24-août-93
10291	24-août-93
10293	26-août-93

N° commande	Date commande
10252	06-juil-93
10253	07-juil-93
10255	09-juil-93
10257	13-juil-93
10259	15-juil-93
10260	16-juil-93
10261	16-juil-93
10263	20-juil-93
10272	30-juil-93
10277	06-août-93
10284	16-août-93
10286	18-août-93
10287	19-août-93
10288	20-août-93
10290	24-août-93
10292	25-août-93

L'intersection $R \cap S$ aura la définition en extension suivante :

N°	Date
10257	13-juil-93
10284	16-août-93
10290	24-août-93



f. Différence

Définition : La différence de deux tables R et S, notée **R - S** ou **MINUS(R,S)**, est une table T de même schéma, contenant les **tuples appartenant à R mais pas à S**

Rappel : deux tables R et S ont le même schéma lorsque les **domaines** associés aux colonnes de R sont **identiques** aux domaines des colonnes de la table S.

Exemple : soient les deux tables qui correspondent à la liste des commandes (Numéro et date) passées avant le 31-juillet-93 et comportant respectivement des produits de catégorie 2 et de catégorie 3 :

R = Commandes de catégorie 2

S = Commandes de catégorie 3

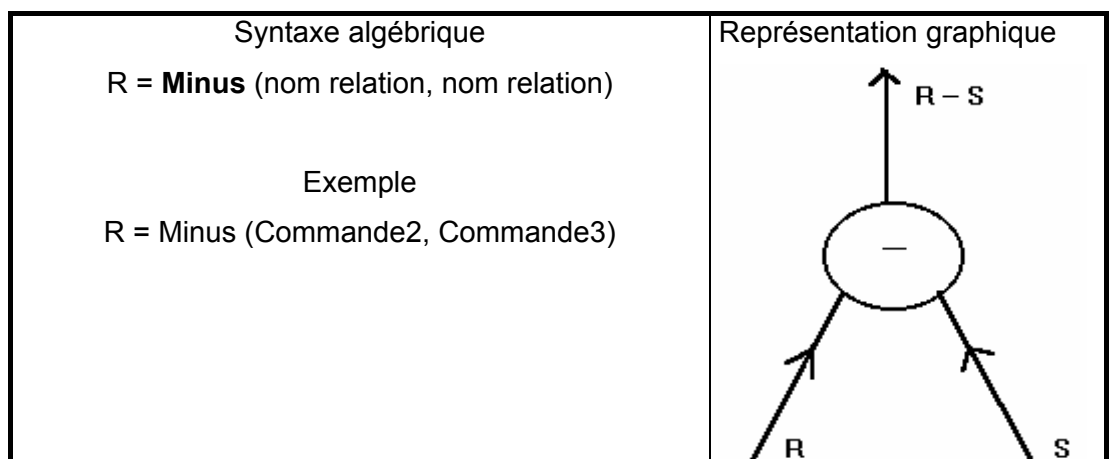
N° commande	Date commande
10250	05-juil-93
10251	05-juil-93
10256	12-juil-93
10257	13-juil-93
10258	14-juil-93
10262	19-juil-93

N° commande	Date commande
10252	06-juil-93
10253	07-juil-93
10255	09-juil-93
10257	13-juil-93
10259	15-juil-93
10260	16-juil-93
10261	16-juil-93
10263	20-juil-93
10272	30-juil-93

La Différence R - S aura la définition en extension suivante :

N° commande	Date commande
10250	05-juil-93
10251	05-juil-93
10256	12-juil-93
10258	14-juil-93
10262	19-juil-93

Interprétation : l'opérateur de différence permet d'exprimer plus facilement des requêtes telles que "**trouver les tuples vérifiant la propriété x mais pas la propriété y**".



g. Composition d'opérations

Avec les opérations de base successivement enchaînées sur des relations il est possible de composer la plupart des questions que l'on peut poser à une base de données relationnelle.

Les questions peuvent donc être exprimées à l'aide de successions des opérations : Union, Intersection, Différence, Division, Jointure, Sélection et Projection.

La représentation graphique de ces opérations permet de composer des **arbres d'opérations relationnelles** :

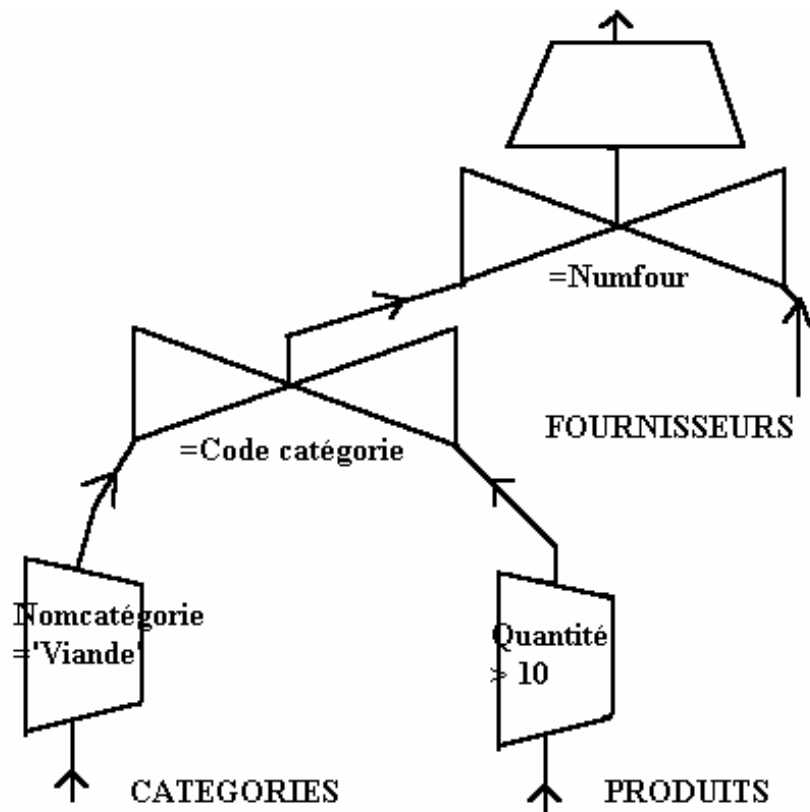
Définition : arbre algébrique

Arbre représentant une requête. Les feuilles sont les relations utilisées dans la requête, chaque noeud intermédiaire est un opérateur algébrique, la racine représente le résultat (arbre organisé de haut en bas) et les arcs représentent les flux de données entre les opérateurs.

Exemple : Dans la base comptoirs à partir des tables Catégories, Produits et Fournisseurs vous devez exprimer la requête :

"Noms des fournisseurs des produits de nom de catégorie = 'Viandes' et ayant une quantité par unité supérieure à 10"

Arbre algébrique basé sur le formalisme de SMITH



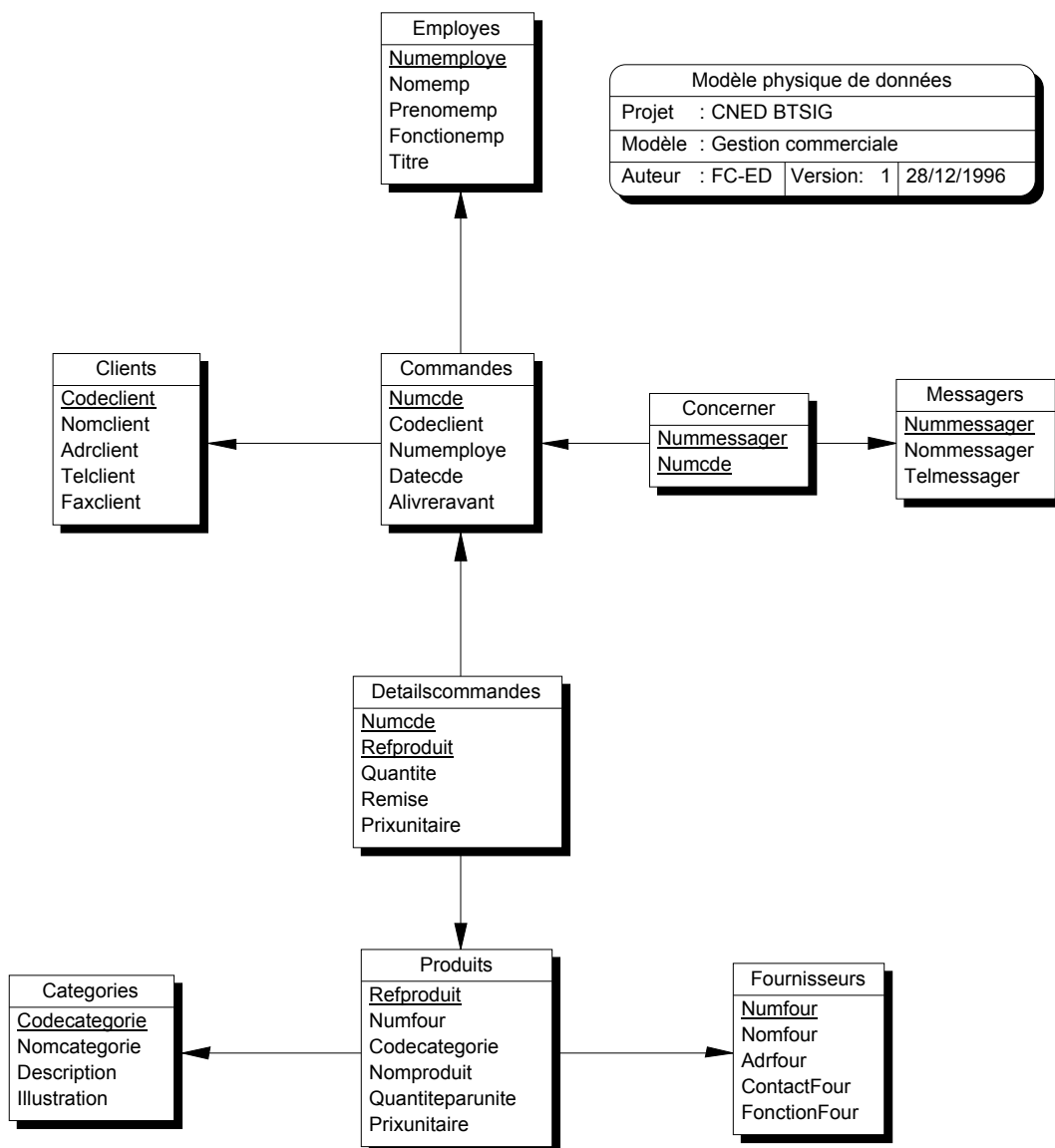
5.C. SQL: un Langage de Manipulation des Données

Remarque. Ce chapitre du cours est une initiation au langage SQL. Il utilise la syntaxe du langage SQL d'Oracle dans sa version 7. Nous avons vu que chaque SGBDR utilise une syntaxe légèrement différente, aussi il se peut que certaines requêtes de ce cours ne fonctionnent pas directement avec un autre SGBDR. Le plus souvent, il suffira de modifier l'écriture de la clause de jointure ou les noms des fonctions pour adapter la requête.

La base COMPTOIR adaptée pour Oracle nous permettra d'utiliser la syntaxe du langage SQL d'Oracle.

Lors de l'écriture des ordres SQL un paramètre entre crochets [] est facultatif.

La codification des noms de tables et des noms de colonnes étant plus restrictive avec Oracle qu'avec Access, nous vous proposons d'utiliser le schéma physique de la base de données ci-dessous :



5.C.1 Projection

Une projection permet d'extraire les colonnes spécifiées d'une table puis élimine les lignes en double.

Une projection s'exprime à l'aide du langage SQL par la clause :

```
SELECT [ALL | DISTINCT] [colonne1 [,colonne2, ...]  
FROM nomtable ;
```

SQL n'élimine pas les doubles à moins que ceci soit explicitement demandé par le mot clé **DISTINCT** ou **UNIQUE** .

Exemple : Liste des clients avec le code client, le nom de la société, l'adresse et le numéro de téléphone s'écrit :

```
SELECT codeclient, nomclient, adrclient, telclient  
FROM clients ;
```

Exemple : Liste des employés avec le nom et la fonction avec élimination des doubles s'écrit :

```
SELECT DISTINCT nomemp, fonctionemp  
FROM employes ;
```

5.C.2 Sélection ou restriction

Une sélection ou restriction s'exprime par un bloc d'instructions du type :

```
SELECT *  
FROM nomtable  
WHERE condition ;
```

* : signifie que l'on affiche toutes les colonnes d'une table.

La combinaison avec une projection s'effectue en remplaçant * par la liste des colonnes à projeter.

Exemple : La liste des produits de la catégorie 3 est obtenue par la requête SQL :

```
SELECT *  
FROM produits  
WHERE codecategorie = 3 ;
```

La même requête avec l'affichage du nom du produit et la quantité par unité s'écrit :

```
SELECT nomproduit, quantiteparunite  
FROM produits  
WHERE codecategorie = 3 ;
```

De manière plus générale, la condition suivant la clause WHERE peut inclure les opérateurs de comparaison :

=	Egal
!=	Différent
>	Supérieur
>=	Supérieur ou Egal
<	Inférieur
<=	Inférieur ou Egal
Between ... AND ...	Entre ... et ...
In	Dans
Like	Comme
Is Null	Est indéfini
Any	Au moins 1
All	Tout

Pour l'opérateur like vous pouvez utiliser les caractères « Joker » suivants :

% : remplace de 0 à n caractères quelconques

like 'M%'

_ : "trait bas" remplace un caractère quelconque et un seul

like '___A%'

Cet opérateur est très utile pour effectuer des recherches dans des chaînes alphanumériques.

Les conditions sont obtenues à l'aide des opérateurs booléens AND, OR, NOT et de parenthèses éventuelles pour indiquer l'ordre d'évaluation.

Il est également possible de trier les résultats suivant l'ordre ascendant (mot clé **ASC**) ou descendant (mot clé **DESC**) d'une ou plusieurs colonnes. Ceci s'effectue en ajoutant une clause du type :

```
ORDER BY {expression | position} [ASC | DESC]  
[, {expression | position} [ASC | DESC], ...]
```

La requête précédente triée par ordre alphabétique sur le nom de produit est effectuée par l'instruction SQL :

```
SELECT nomproduit, quantiteparunite  
FROM produits  
WHERE codecategorie = 3 ;  
ORDER BY nomproduit ;
```

Une autre syntaxe possible consiste à indiquer la position de la colonne dans la clause Select.

```
SELECT nomproduit, quantiteparunite  
FROM produits  
WHERE codecategorie = 3 ;  
ORDER BY 2 desc ;
```

permet d'obtenir la liste des produits de la catégorie 3, triée par quantité par unité décroissante.

Remarque : la clause Distinct réalise un tri sur les colonnes et élimine les valeurs dupliquées issues de l'interrogation.

5.C.3 Jointures

Un cas particulier simple de **jointure sans condition** est le produit cartésien.

Définition : Le produit cartésien, de deux tables R et S, est une table T ayant pour attributs la concaténation des attributs de R et S. Tous les tuples de T sont obtenus par la concaténation de chaque tuple de R à tous les tuples de S.

Avec SQL, le produit cartésien s'exprime en incluant plusieurs tables dans la clause FROM.

Exemple : Produit Cartésien des tables Clients et Commandes

```
SELECT *  
FROM Clients, Commandes ;
```

La **jointure avec condition** peut s'exprimer de plusieurs manières. Une première expression naturelle est la restriction du produit cartésien.

Exemple : La jointure de Clients et Commandes sur la colonne code client peut s'écrire :

```
SELECT *  
FROM Clients, Commandes  
WHERE Clients.codeclient = Commandes.codeclient ;
```

L'écriture ci-dessus correspond à une **formulation prédicative**.

Le préfixage de la colonne Codeclient par le nom de la table est obligatoire car cette colonne a la même appellation dans les tables Clients et Commandes.

La jointure peut aussi être exprimée d'une manière plus procédurale avec des blocs imbriqués reliés par l'opérateur **IN** :

Liste des clients ayant commandés	Liste des commandes associées à un client
<pre>SELECT * FROM Clients WHERE codeclient IN (SELECT codeClient FROM Commandes) ;</pre>	<pre>SELECT * FROM Commandes WHERE codeclient IN (SELECT codeClient FROM Clients) ;</pre>

L'écriture ci-dessus correspond à une **formulation algébrique**.

IN peut être vu comme l'opérateur ensembliste d'appartenance (\in)

Cette dernière formulation ne permet d'afficher que des informations provenant d'une seule table.

5.C.4 Combinaison de la sélection, de la jointure et de la projection

Avec le langage SQL, vous pouvez très facilement combiner les opérations de jointure, sélection et projection. Il est également possible d'imbriquer des blocs SELECT .. FROM .. WHERE à plusieurs niveaux.

Exemple : "Noms des fournisseurs des produits de nom de catégorie = Viandes. "

Formulation prédicative :

```
SELECT DISTINCT nomfour
FROM fournisseurs, produits, categories
WHERE categories.nomcategorie = "Viande"
      AND categories.codecategorie = produits.codecategorie
      AND fournisseur.numfour = produits.numfour ;
```

Formulation algébrique :

```
SELECT DISTINCT nomfour
FROM fournisseurs
WHERE numfour IN
  ( SELECT numfour
    FROM produits
    WHERE codecategorie IN
      ( SELECT codecategorie
        FROM categories
        WHERE nomcategorie="Viandes" ) ) ;
```

Formulation mixte :

```
SELECT DISTINCT nomfour
FROM fournisseurs
WHERE numfour IN
  ( SELECT numfour
    FROM produits, categories
    WHERE categories.codecatégorie = produits.codecatégorie
      AND nomcategorie="Viandes" );
```

Une forme particulière de l'opération de jointure consiste à n'afficher qu'une des colonnes communes aux deux tables. Il s'agit alors d'une **jointure naturelle**.

```
SELECT Commandes.*, nomclient
FROM   Clients, Commandes
WHERE  Clients.Codeclient = Commandes.Codeclient ;
```

5.C.5 Union, Intersection et Différence

Les opérateurs ensemblistes disponibles avec Oracle reposent sur la syntaxe et les règles suivantes :

Syntaxe :

```
Select ...
      Opérateur ensembliste
Select ...
```

Règles :

1. Même nombre de variables, de types identiques dans chacun des select. Ceci correspond à la définition de l'unicompatibilité des schémas de relation.
2. Les doublons sont éliminés (distinct implicite)
3. Les titres des colonnes sont identiques à ceux du premier Select.
4. La largeur de la colonne est égale à la plus grande de tous les select.
5. La clause Order By fait référence au numéro de colonne et non plus à son nom.

L'opérateur d'union est UNION.

L'opération d'union permet de juxtaposer des tables verticalement.

Exemple : Liste des tiers de l'entreprise

```
SELECT codeclient, nomclient
FROM   clients
UNION
SELECT numfour, nomfour
FROM   fournisseurs ;
```

Avec Oracle Version 7, vous disposez de l'opérateur : **UNION ALL**

L'opérateur d'intersection est : **INTERSECT**

Exemple : Liste des commandes (Numéro et date) comportant des produits de catégorie 2 et de catégories 3.

```
SELECT numcde, datecde
FROM  Commandes, DetailsCommandes, Produits
WHERE codecategorie=2
and Commandes.numcde = DetailsCommandes.numcde
and DetailsCommandes.Refproduit = Produits.Refproduit
INTERSECT
SELECT numCde, datecde
FROM  Commandes, DetailsCommandes, Produits
WHERE codecategorie=3
and Commandes.numcde = DetailsCommandes.numcde
and DetailsCommandes.refproduit = Produits.refproduit
;
```

L'opérateur de différence est : **MINUS**

Exemple : Liste des commandes (Numéro et date) comportant des produits de la catégorie 2 sans comporter des produits de la catégorie 3.

```
SELECT numcde, datecde
FROM  Commandes, DetailsCommandes, Produits
WHERE codecategorie=2
and Commandes.numcde = DetailsCommandes.numcde
and DetailsCommandes.refproduit = Produits.refproduit
MINUS
SELECT numcde, datecde
FROM  Commandes, DetailsCommandes, Produits
WHERE codecategorie=3
and Commandes.numcde = DetailsCommandes.numcde
and DetailsCommandes.refproduit = Produits.refproduit
```

5.C.6 Les Expressions et les Fonctions

Une expression est une combinaison de variables, de constantes, de fonctions au moyen d'opérateurs.

Une fonction est une routine avec des valeurs dépendantes de leurs arguments qui peuvent être des fonctions.

Les expressions et les fonctions peuvent être utilisées dans les clauses Select, Where, Order By et Group By.

a. Les fonctions agrégatives

Elles permettent d'effectuer des calculs "verticaux". Par exemple sur l'ensemble ou un sous-ensemble des valeurs d'une colonne. Les fonctions disponibles sont :

SUM : permet d'effectuer la somme des valeurs d'une colonne numérique

AVG : permet d'effectuer la moyenne des valeurs d'une colonne numérique

MAX : permet de rechercher la valeur maximale d'une colonne numérique

MIN : permet de rechercher la valeur minimale d'une colonne numérique

COUNT : permet de compter le nombre de valeur d'une colonne

- pour compter le nombre de lignes sélectionnées, la fonction doit être utilisée avec l'argument *

- pour compter le nombre de valeurs différentes prises par une colonne, il faut indiquer l'argument **DISTINCT** suivi de l'argument considéré.

VARIANCE : permet de calculer la variance d'une colonne numérique

STDDEV : permet de calculer l'écart-type d'une colonne numérique

Toutes ces fonctions peuvent être utilisées avec les options **DISTINCT** ou **ALL** ; (par défaut ALL)

Exemple : "Calcul du prix moyen des produits commandés"

```
SELECT AVG (prixunitaire)
FROM Detailscommandes ;
```

b. Expressions arithmétiques

Il est possible d'effectuer dans une requête SQL, des calculs "horizontaux" sur des lignes, en utilisant les opérateurs :

+, -, *, /

aussi bien dans la clause SELECT que dans un WHERE (expression d'une condition sur des valeurs calculées). Les arguments des opérateurs sont des noms de colonnes de type numérique ou des constantes.

Exemple : "Quels sont les lignes de commandes qui avec une augmentation de la quantité de 10 % ont une quantité supérieure à 100 ?"

```
SELECT *  
FROM Detailscommandes  
WHERE Quantite * 1.1 > 100 ;
```

Exemple : "Calcul du montant non remisé de la comande numéro 10248"

```
SELECT SUM (PrixUnitaire*Quantité)  
FROM Detailscommandes  
WHERE Numcde = 10248 ;
```

Pour les calculs on peut utiliser :

- une colonne (+, -, *, /) une colonne
- une constante (+, -, *, /) une constante
- une fonction arithmétique (+, -, *, /) une fonction arithmétique

c. Expressions chaîne de caractères

L'opérateur de concaténation symbolisé (||) par la double barre verticale (Alt GR 6 sur votre clavier) est disponible avec Oracle. Une fonction Concat est disponible avec Oracle 7.2

Exemple : "Liste des noms et adresse des clients sur une seule colonne."

```
SELECT nomclient || '→' || adrclient  
FROM clients ;
```

d. Expressions sur les dates

SYSDATE : variable représentant la date et l'heure du système

```
Select SYSDATE from dual;
```

Les opérateurs à votre disposition sont : + et -

Exemples

Date1 + Nombre = Date2

'10-OCT-96' + 3 = '13-OCT-96'

Date1 - Nombre = Date2

'13-OCT-96' - 3 = '10-OCT-96'

Date1 - Date2 = Nombre

'13-OCT-96' - '10-OCT-96' = 3

e. Les fonctions

Pour faciliter les calculs il existe des fonctions appliquées aux chiffres.

1. les fonctions arithmétiques

ABS (n) : Valeur Absolue de n

ABS(-23.5) = 23.5

MOD (M,N) : Division entière de M par N

MOD(7,2) = 1

SIGN (n) : signe d'un entier : -1 si n<0, 0 si n=0, 1 si n>0

SIGN(-23.5) = -1

SIGN(0) = 0

SIGN(+23.5) = +1

POWER (M,N) : puissance

POWER(2,3) = 8

SQRT (n) : racine carrée de n

SQRT(16) = 4

SQRT(-16) = NULL

ROUND(N,[M]) : arrondi d'un nombre

ROUND(23.462,2)=23.46

ROUND(23.466,2)=23.47

ROUND(1200,-3)=1000

ROUND(1500,-3)=2000

TRUNC(N,[M]): troncature d'un nombre (partie entière si M=0)

TRUNC(23.462,2)=23.46

TRUNC(23.466,2)=23.46

TRUNC(1200,-3)=1000

La plupart des langages SQL disposent également de fonctions appliquées aux caractères, de fonctions de conversion et de fonctions basées sur les différents formats de date.

L'opérateur de concaténation : || permet de présenter deux colonnes sur une seule colonne.

2. les fonctions pour les caractères

CHR (n) : retourne le caractère de la valeur ascii n

CHR(65)='A'

INITCAP (chaîne) : première lettre en majuscule, reste minuscule

INITCAP('dupont')='Dupont'

LENGTH (chaîne) : longueur d'une chaîne de caractères

LENGTH('DUPONT')=6

LPAD(chaîne,longueur,caractère) = compléter par la gauche

LPAD('Dupont',10,'*')='*****Dupont'

LOWER (chaîne) : transformation en minuscule

LOWER('Dupont')='dupont'

RPAD(chaîne,longueur,caractère) = compléter par la droite

RPAD('Dupont',10,'*')='Dupont*****'

SOUNDEX (chaîne) : fonction de comparaison phonétique

SOUNDEX(nompilote)=SOUNDEX('Dupont')

Génère une valeur alphanumérique sur 4 octets en appliquant les règles suivantes :

minuscule=majuscule,

lettre doublée = lettre simple,

voyelles éliminées (sauf si première lettre)

consonnes → valeur de 1 à 6 (1 = B,F,P,V 2= C,G,J,K,Q,S,X,Z 3= ...6 =R)

SUBSTR (chaîne, position (,longueur) : extraire une chaîne

SUBSTR('MARTIN',2,3) = 'ART'

UPPER (chaîne) : transformation en majuscule

UPPER('Dupont') = 'DUPONT'

5.C.7 Le partitionnement

Le partitionnement doit permettre d'effectuer des calculs statistiques par ensemble de lignes vérifiant un même critère.

Le partitionnement s'exprime en SQL par la clause **GROUP BY**, suivie du nom des colonnes de partitionnement.

Sur certains SGBD, lorsque vous utilisez dans une requête SQL le partitionnement vous devez impérativement utiliser une fonction agrégative dans la clause SELECT. (pas Oracle)

Lorsqu'une condition de recherche doit être exprimée non pas sur les lignes d'une table mais sur les classes de lignes introduites par un partitionnement, la clause à utiliser est **HAVING**.

Exemple : "Rechercher toutes les commandes comportant plus de 10 lignes de détails"

```
SELECT numcde
FROM Detailscommandes
GROUP BY Numcde HAVING COUNT (*) > 10 ;
```

Incohérence entre le partitionnement et la projection :

```
SELECT Numcde, prixunitaire
FROM Detailscommandes
GROUP BY Numcde HAVING COUNT (*) > 10 ;
```

La requête ci-dessus génère un message d'erreur car elle risque de créer un tableau dont certains champs seraient amenés à accueillir une liste de valeurs (champs multivalués).

En effet, dans notre exemple, il y a plusieurs prix unitaire pour une commande. On est donc obligé de modifier la présentation de l'information en appliquant une fonction agrégative aux colonnes contenant des champs multivalués.

Vous pouvez combiner dans une requête la sélection des lignes (clause Where) avec la sélection des groupes (clause Having).

L'**évaluation de la requête** s'effectue dans l'ordre suivant :

- 1.Evaluation de la clause Where
- 2.Evaluation de la clause Group by
- 3.Evaluation des fonctions agrégatives
- 4.Evaluation de la clause Having
- 5.Evaluation de la projection

5.C.8 Les Mises à jour

a. Insertion de données

L'insertion des lignes dans une table peut être :

1. "Totale"

C'est-à-dire qu'il y a affectation de valeur pour toutes les colonnes de la table.

Les valeurs doivent être saisies entre "(" et ")" , séparées par des "," en suivant impérativement l'ordre de définition des colonnes de la table.

INSERT INTO nomtable VALUES (Valeur1, ..., Valeurn) ;
--

Exemple : Insérer la ligne :

(10, 'Liess','Jean Charles','Directeur Informatique','M.') dans Employés

```
INSERT INTO EMPLOYES
```

```
VALUES(10, 'Liess','Jean Charles','Directeur Informatique','M.');
```

2. "Partielle"

Seules les colonnes nommées sont valorisés.

Les valeurs saisies doivent suivre l'ordre dans lequel apparaissent les colonnes. Pour les colonnes non citées, il y a affectation automatique de valeur nulle.

Remarque : l'insertion partielle exige la valorisation de toutes les colonnes définies en NOT Null.

INSERT INTO nomtable (colonne1,... ,colonnen) VALUES (valeur1,... ,valeurn);
--

Exemple :

Insérer la ligne : (9, 'Bricolage','Outillages, Electricité, Peinture') dans la table catégories

```
INSERT INTO CATEGORIES (codecategorie, nomcategorie, description)
```

```
VALUES (9, 'Bricolage','Outillages, Electricité, Peinture');
```

b. Suppression de données

L'opération de suppression de lignes dans une table est effectuée pour toutes les lignes vérifiant la condition indiquée.

L'instruction SQL de suppression est la suivante :

```
DELETE FROM nomtable  
[WHERE condition ];
```

Remarque :

Si la clause Where n'est pas utilisée, la commande Delete efface toutes les données de la table.

Il faut également attention à l'intégrité référentielle qui peut interdire ou propager des suppressions de lignes.

Exemple : Supprimer toutes les lignes de détails de la commande 10248

```
DELETE FROM Detailscommandes  
WHERE numcde= 10248;
```

c. Modification des données

L'opération de modification dans la table spécifiée est effectuée pour toutes les lignes vérifiant la condition indiquée.

En l'absence de sélection, les modifications sont apportées sur toutes les lignes de la table.

L'attribution de nouvelles valeurs est effectuée dans la clause **SET**.

```
UPDATE nomtable  
SET { colonne = expression [,colonne = ]  
      [(colonne [,colonne,...]) = (requête) }  
[WHERE condition ] ;
```

Cette syntaxe permet plusieurs types de modifications de données.

Exemple : tous vos clients sont dans la même zone téléphonique (03), suite à la mise en place de la nouvelle numérotation vous devez mettre à jour le numéro de téléphone et le numéro de fax de tous vos clients.

```
UPDATE clients  
SET telclient = '03' || telclient,  
    faxclient = '03' || faxclient ;
```

Module 3 **Méthodes de conception de systèmes d'informations**

1. PRÉSENTATION

Cette module présente la notion de projet informatique, puis les caractéristiques générales de la méthode de conception de systèmes d'information : MERISE.

Ensuite nous aborderons la modélisation conceptuelle des données, telle est préconisée dans la méthode Merise, en étudiant les modèles de communication et de traitements.

1.A. Références techniques

AMC*Designor Module de Traitements a été utilisé pour réaliser les différents modèles de données de cette séquence.

1.B. Références bibliographiques

- Pour la pratique de la modélisation utilisez AMC*Designor 5.1 version évaluation livré sur CD-ROM avec le livre :

AMC*Designor : Mise en œuvre de Merise et Conception d'applications client-serveur par Gilles GUEDJ aux éditions EYROLLES - parution Octobre 96.

2. RÉALISATION D'UN PROJET INFORMATIQUE

La démarche classique conduisant à la réalisation d'un projet informatique est généralement découpée en étapes :

Ces étapes constituent l'analyse informatique. Des méthodes d'analyse basées sur l'expérience ont été conçues pour assister l'analyste-programmeur chargé de réaliser un projet informatique.

2.A. Qu'est-ce qu'une analyse informatique ?

L'analyse au sens informatique du terme consiste à :

- Comprendre et modéliser la gestion du domaine d'activité dans une organisation (entreprise ou administration)
- Concevoir la solution informatique adéquate

L'efficacité et la validité d'une analyse reposent sur la qualité de la communication entre les demandeurs de services (utilisateurs) et les fournisseurs de services (informaticiens).

2.A.1 Définition

L'analyse informatique a pour but de définir tous les éléments organisationnels, logiciels et matériels nécessaires à la gestion automatisée d'une organisation.

2.A.2 Exemple en entreprise

Exemple de travail d'analyse : mettre en place un système de gestion des réservations de chambres dans un hôtel. Comment faire ? Il s'agit d'un travail de conception. On sera amené inévitablement à se poser les questions suivantes :

Quelle est l'organisation actuelle ? Comment ça marche ? Qu'est ce qui ne fonctionne pas bien ? Pourquoi ? Quelles sont les informations manipulées ? par qui ? quand ? Comment va t-on les stocker ? Comment va t-on y accéder ? Quels traitements sont réalisés ? Quand ? Comment ? Par qui ? Quels programmes sont à écrire ?

L'objectif du cours d'analyse est de donner des méthodes de travail pour concevoir un système de gestion d'informations cohérent, complet et fiable en répondant à toutes ces questions.

2.B. Système d'information (SI)

2.B.1 Définition d'un système d'information

Toute entité qui rassemble des moyens organisés pour atteindre un but est un "système". Un système peut toujours être décomposé en trois sous-systèmes : Le système *opérant*, le système de *décision* et le système *d'information*. Ex du corps humain (muscles puis cerveau et enfin système nerveux).

L'analyse informatique s'intéresse au système d'information en tant que support des données et des traitements gérés par une entreprise. Une analyse informatique doit aboutir à une définition détaillée des outils de

gestion (informatique ou autre) qui permettront d'atteindre les objectifs de l'entreprise.

Une analyse concerne rarement une entreprise dans sa totalité mais plutôt un service particulier ou une fonction précise. On parle du domaine de l'analyse.

2.B.2 La démarche de l'analyste

- Identifier le domaine à étudier : lieux, personnes, matériels, fonctions.
- Comprendre et décrire le fonctionnement actuel du système dans le domaine étudié : objectifs, données gérées, traitements réalisés, postes de travail, circulation et stockage de l'information.
- Faire une analyse critique de l'existant.
- Recenser les nouveaux besoins de gestion.
- Proposer une ou plusieurs solutions pour le fonctionnement du futur système.
- Définir en détails le fonctionnement du futur système.
- Suivre sa mise en oeuvre.

Exemple : architecte.

2.B.3 Exemples

- Un plan d'architecte "modélise" le bâtiment réel existant ou futur.
- On parle de modèle mathématique pour décrire le comportement d'un phénomène physique. Le modèle décrit de façon théorique le phénomène ce qui permet de prévoir son comportement. La réalité amène toujours quelques écarts par rapport au modèle. Ex : la météo.
- Un circuit logique "modélise" un composant électronique.

2.C. Méthodes d'analyse

Les méthodes d'analyse et de conception intègrent, à des degrés divers, les trois éléments ci-dessous :

2.C.1 La Démarche

La **démarche** est le processus opératoire qui permet d'effectuer le travail de modélisation de description et de réalisation d'un système informatique

2.C.2 Les modèles

Un système d'information réel est souvent très complexe. Pour mieux le comprendre et analyser son comportement il sera représenté sous forme de modèles. On parle de modélisation d'un système. Un modèle est une représentation simplifiée et normalisée de tout ou partie d'un système d'information.

Un modèle peut être vu à la fois comme un outil de recherche d'une solution optimale et comme un outil de représentation de cette solution.

Le plus souvent, les modèles sont des graphiques qui représentent un aspect d'un système d'information tel qu'il est perçu par l'analyste. Attention : le mot

"modèle" n'est pas pris dans le sens "exemple qui fait référence" mais plutôt comme un "moule" dans lequel doit rentrer la description du réel tel qu'il est perçu par l'analyste.

2.C.3 Les outils

Les outils logiciels, nommés généralement Atelier de Génie Logiciel (AGL) supportent une méthode d'analyse en facilitant la conception, la documentation, la simulation, l'évaluation ou la réalisation du système d'information.

2.D. LA DÉMARCHE DE CONCEPTION

Tout projet nécessite une spécification rigoureuse du besoin informatique.

Le cahier des charges décrit le POURQUOI de la situation qui occasionne le problème.

La spécification du problème décrit le QUOI, c'est-à-dire les caractéristiques externes complètes de la solution à concevoir, qui devra fonctionner dans l'environnement précisé dans le cahier des charges.

La conception décrit le COMMENT, la manière dont la solution présentée propose de résoudre le problème spécifié.

Dans cette démarche, la modélisation est le processus qui permet de réaliser une représentation approchée et simplifiée de la réalité étudiée.

Le modèle peut être considéré comme l'outil méthodologique permettant de décrire, analyser, définir, concevoir en utilisant une démarche normalisée. On peut distinguer plusieurs types d'approche de la modélisation :

- approche par les traitements,
- approche par les données,
- approche par les événements extérieurs,
- approche par les résultats.

Ces approches sont complémentaires et s'appliquent aux différentes étapes du cycle de développement.

2.E. LE DÉVELOPPEMENT

L'objectif d'un programmeur est de concevoir et de coder des programmes qui répondent au problème posé, qui soient clairs, structurés, fiables, performants, documentés, faciles à lire, à comprendre et à maintenir.

Il doit, pour ce faire, utiliser un ensemble de techniques, de procédures, de standards visant à une meilleure qualité, à une meilleure efficacité et à un meilleur contrôle.

Les principes en sont :

- La conception des programmes par analyse descendante (*Top-Down*) qui présente l'avantage de permettre une validation progressive, une intégration continue et des interfaces documentées ;
- La codification des programmes selon les principes de la programmation structurée, en utilisant un pseudo-code ;

- la documentation des programmes en utilisant des bibliothèques de développement qui permettent de disposer d'une représentation à jour du produit développé et de fournir une base de travail ordonnée pour le contrôle, et ce grâce à un responsable ainsi qu'à des outils de maintenance des versions et des documentations ;
- Une organisation en équipes de développement de 3 à 5 personnes autour d'un chef de groupe ;
- L'organisation de sessions de révision de code. Organisée par l'auteur des programmes, elles permettent de passer en revue le travail effectué, de découvrir et de corriger les erreurs le plus tôt possible dans le cycle de développement.

2.F. LES TESTS DE LOGICIEL

La validation d'un logiciel s'effectue grâce à des tests.

Ils ne permettent pas d'établir qu'un logiciel est correct, ni de mettre en évidence l'absence de défauts mais au contraire de démontrer la présence de dysfonctionnement.

3. LA MÉTHODE D'ANALYSE MERISE

3.A. Présentation

MERISE est une méthode d'analyse apparue en 1979 et conçue conjointement par des industriels et des universitaires. C'est la méthode la plus connue aujourd'hui en France. Elle est utilisée par de grandes sociétés de services informatiques dont CAP SESA, CGI, GAMMA, SEMAGROUP.

La méthode MERISE propose une démarche pour définir un système d'information. Cette démarche amène l'analyste à concevoir des modèles représentatifs du système d'information actuel et futur. Ce cours donne une vue partielle de la méthode.

3.B. Caractéristiques

Merise est une méthode de conception et de développement de systèmes d'information, elle se caractérise par une double démarche, par niveaux et par étapes. Merise propose une approche de conception séparant l'étude des données de celle des traitements, en avançant progressivement par niveaux.

L'analyse des données donne une vision statique du domaine étudié alors que l'analyse des traitements s'intéresse à l'aspect dynamique du système.

La séparation entre données et traitements assure cohérence et pérennité au système.

- Cohérence : le système est étudié dans sa globalité.
- Pérennité : L'apparition de nouveaux besoins de traitements ne remettra pas en cause l'organisation des données.

Exemple :

Pour calculer le prix TTC d'un produit il faut connaître le taux de TVA, or, si ce dernier ne fait pas partie des données mais des traitements (valeur mémorisée dans le programme), quand le taux de TVA change, il faut également changer le programme. Une analyse des données avisée aura prévu la mémorisation du taux de TVA sans tenir compte des traitements.

La démarche par niveaux a pour objectif la formalisation du futur système sous ses différents aspects (contribution à la stratégie d'entreprise, mise en oeuvre des règles de gestion, aspects organisationnels et techniques).

Cette démarche part du théorique et va vers le pratique.

Au **niveau conceptuel** on applique un certain nombre de méthodes théoriques pour aboutir à un modèle idéal.

Au **niveau logique** on adapte ce modèle idéal à différentes contraintes techniques : environnement informatique effectivement disponible, temps de réponse, espace de stockage disponible...

Au **niveau physique** on sort du cadre de l'analyse pour mettre effectivement en place des fichiers, des bases de données et des programmes.

3.C. Les modèles de MERISE

3.C.1 Présentation

La méthode Merise propose par niveaux les modèles suivants :

Niveau de Modélisation	Choix	Préoccupations	Modèles de Données	Modèles de Traitements
Conceptuel	Gestion	Quoi ?	MCD	MCT
		Que veut-on faire ?		
Logique	Organisation	Qui fait quoi ? Ou ?	MLD	MOT
		Quant ? Comment ?		
Physique	Techniques	Avec quels moyens ?	MPD	MOpT

Données	Traitements
<u>MCD</u> : Modèle conceptuel des données	<u>MCT</u> : Modèle conceptuel des traitements
<u>MLD</u> : Modèle logique des données	<u>MOT</u> : Modèle organisationnel des traitements
<u>MPD</u> : Modèle physique des données	<u>MOPT</u> : Modèle opérationnel des traitements

3.C.2 Rôle

Les modèles servent à répondre aux questions suivantes :

Données	Traitements
MCD : Quelles données et quels liens entre les données ?	MCT : Quels traitements ?
MLD : Comment organiser le stockage données ?	MOT : Comment organiser les traitements ?
MPD : Comment stocker effectivement les données ?	MOPT : Comment réaliser les traitements ?

3.D. Le suivi d'un projet MERISE

Dans la méthode MERISE on parle de PROJET pour identifier l'analyse d'un Système d'Information particulier.

La démarche par étapes a pour objectif la hiérarchisation des décisions qui doivent être prises au cours de la conception, du développement, de la mise en oeuvre du nouveau système d'information, mais aussi de l'évolution du système qui sera mis en place.

Les différentes étapes sont les suivantes :

3.D.1 Schéma directeur

Le schéma directeur est une étape de réflexion globale et de planification du développement du système d'information. L'objectif de cette première étape est de faire le pont entre la stratégie de l'entreprise et ses besoins en termes de systèmes d'information.

Pour ce faire, l'entreprise sera décomposée en domaines de gestion.

En parallèle, il sera procédé à l'identification des activités de l'entreprise, qui s'expriment habituellement en termes de finalités stratégiques. Voici quelques exemples : concevoir des produits nouveaux, les vendre, acheter des matières premières, gérer le personnel.

Une fois les domaines de gestion et les finalités identifiés, il s'agira d'affecter les finalités aux domaines correspondants. Ceci étant fait, on pourra procéder à la recherche des facteurs critiques de succès.

Connaissant les facteurs critiques de succès du domaine, on pourra dès lors procéder à l'identification des besoins en termes de systèmes d'information pour le domaine en question.

C'est à ce niveau que sont définies les priorités concernant les différentes applications à développer, les budgets prévisionnels mais aussi le niveau de fiabilité et de confidentialité de chacune d'elles et que sont effectués divers arbitrages. Le résultat obtenu à l'issue de cette étape est un Plan Directeur Informatique.

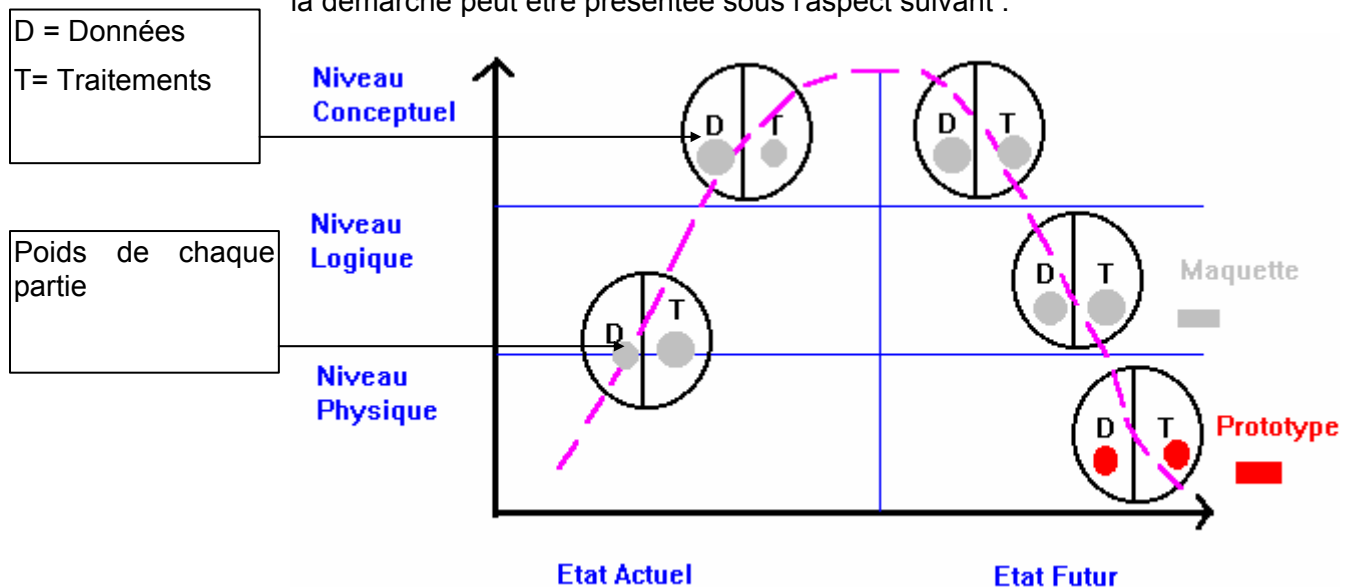
3.D.2 Etude préalable

A la suite du schéma directeur, l'étude préalable aura pour but de reprendre domaine par domaine et d'étudier de manière plus approfondie les projets à mettre en oeuvre et leur interfaçage. Son objectif est la constitution d'un Dossier de Choix permettant d'apprécier les diverses solutions alternatives d'informatisation, pour un domaine donné en tenant compte des orientations définies par le schéma directeur.

Faire une étude préalable, c'est aussi se donner la possibilité d'actualiser, le cas échéant, un certain nombre de détails du schéma directeur et de vérifier l'opportunité des projets identifiés au niveau des domaines.

L'étude préalable doit définir les principales solutions envisageables et pour chacune d'elles, elle doit préciser ses avantages et ses inconvénients, les coûts et les moyens ainsi que les délais prévus pour l'étude détaillée, la production, la mise en oeuvre et la maintenance.

La schématisation des influences de cette nouvelle étape sur les modèles de la démarche peut être présentée sous l'aspect suivant :



- Une maquette est un ensemble logiciel ayant vocation à démontrer la faisabilité d'une application de gestion ou à en illustrer les fonctionnalités de point de vue externe, celui de l'utilisateur final. La technique de développement utilisée n'est pas représentative de la technique réelle envisagée.
- Un prototype quant à lui est un logiciel intégrant tout ou partie des fonctionnalités réelles à développer mais n'utilisant qu'un sous-ensemble des données réelles d'entrée. Le point de vue pris en compte est le point de vue interne, celui du réalisateur. La technique de développement employée est celle qui le sera pour l'ensemble applicatif final et le prototype est susceptible d'être utilisée dans des conditions réelles, sur un site pilote par exemple, avant généralisation. L'utilisation possible de prototype renforce la représentativité de l'étude préalable.

3.D.3 Etude détaillée

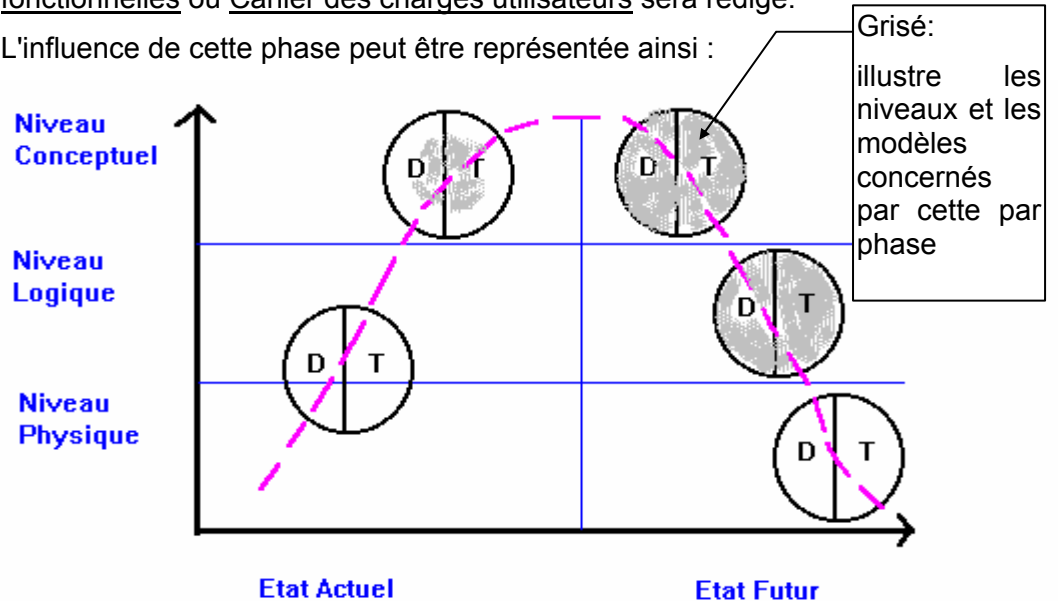
Elle complète l'étude préalable et plus spécifiquement la solution qui a été choisie parmi celles ayant fait l'objet de l'étude préalable.

L'étude détaillée est effectuée projet par projet, ne porte que sur les fonctions à automatiser dans le cadre d'un même projet et se décompose en deux phases principales :

1. la conception fonctionnelle,
2. la conception ou étude technique.

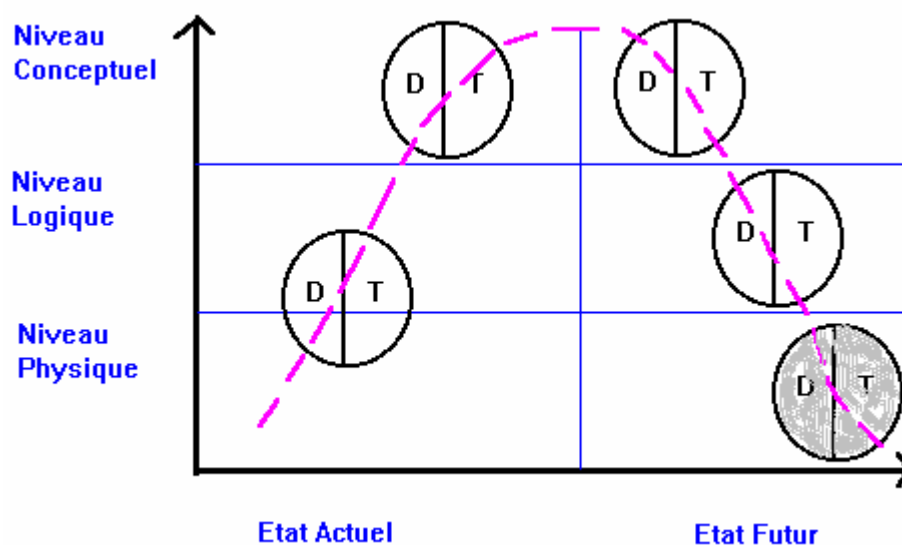
A l'issue de la conception fonctionnelle un Dossier de spécifications fonctionnelles ou Cahier des charges utilisateurs sera rédigé.

L'influence de cette phase peut être représentée ainsi :



L'étude technique aboutit à la rédaction d'un Dossier de spécification techniques ou Cahier de charges de réalisation, dossier indispensable pour la production du logiciel.

Le schéma prend alors l'allure suivante :



3.D.4 Réalisation

Elle consiste en une fabrication des différents programmes et transactions dont l'architecture a été définie, ainsi qu'en une fabrication des structures de fichiers ou bases de données qui seront ultérieurement remplis par les utilisateurs.

La réalisation est effectuée en trois parties :

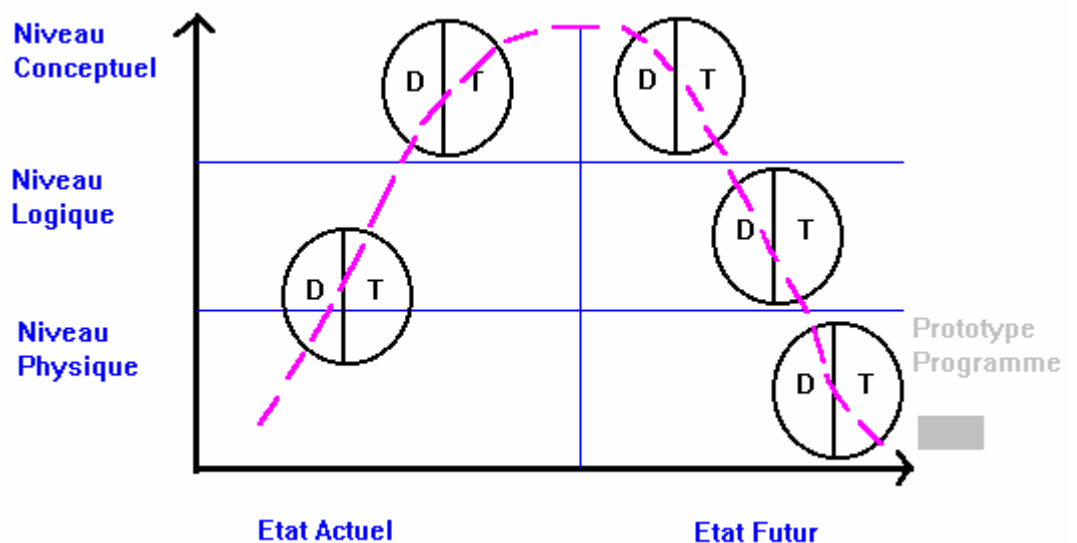
- codage des programmes,
- tests et mise au point,
- intégration de l'ensemble des transactions et des chaînes batch (ou traitement par lots qui concerne un ensemble de données souvent réalisé en différé).

Les logiciels sont réalisés de manière modulaire ou structurée et donnent lieu à des tests unitaires, puis sont regroupés pour donner à des tests d'intégration. Ces derniers sont effectués sur les jeux d'essais utilisateurs. Cette production peut passer par la fabrication d'un prototype.

Les difficultés relatives à cette étape sont bien connues des informaticiens. Traditionnellement, on prend ici autant sinon plus de temps à mettre au point les programmes et à les intégrer qu'on en a pris pour effectuer la phase de conception et d'analyse.

L'étape est sanctionnée par une Documentation technique d'exploitation.

L'influence sur les modèles est la suivante :



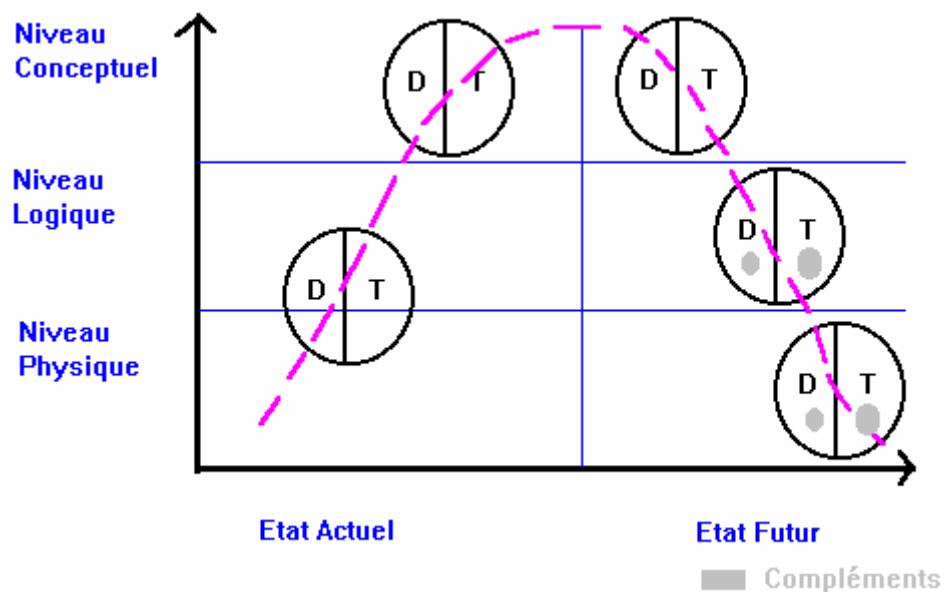
3.D.5 Mise en œuvre et généralisation

La mise en œuvre des nouvelles applications nécessite également un temps non négligeable. Imaginons en effet une compagnie d'assurance qui entre en machine tous ses assurés : cela ne peut se faire en une semaine si la compagnie a un million d'assurés.

Ainsi la mise en œuvre des applications nécessite la réalisation de l'ensemble des tâches suivantes :

- création et initialisation des bases de données,
- réception éventuelle et installation des nouveaux matériels informatiques,
- rédaction de manuels pour les futurs utilisateurs,
- formation des futurs utilisateurs aux nouvelles applications,
- lancement des nouvelles applications en parallèle avec les anciennes,
- lancement définitif des nouvelles applications.

C'est pourquoi l'étape de mise en œuvre peut commencer très tôt dans le processus de conception, puisqu'elle peut débuter dès le stade de l'étude préalable :



Elle se concrétise par la mise à disposition des utilisateurs du système développé, que celui-ci soit à l'état de prototype ou complet. La recette de ce qui a été développé se fait sur la base de jeux d'essais spécifiques. Cette recette est à ce stade provisoire.

Afin d'apporter des garanties de bon fonctionnement, cette recette provisoire ne peut devenir définitive qu'à l'issue d'une période d'essais intensifs sur la base de la montée en charge réelle du système nouvellement livré. Le Protocole de recette est le document qui sanctionne la mise en œuvre.

La généralisation consiste à mettre le système à disposition de l'ensemble des utilisateurs, en intégrant toute la variété des ressources et des contextes organisationnels dans lesquels le dit système doit pouvoir être exploité. Elle sera utilisée dans le cadre d'une entreprise de taille importante ayant par exemple des matériels d'exploitation différents et pour lesquels une adaptation du système sera nécessaire.

Un Dossier de mise en exploitation de chacune des versions clôt cette étape.

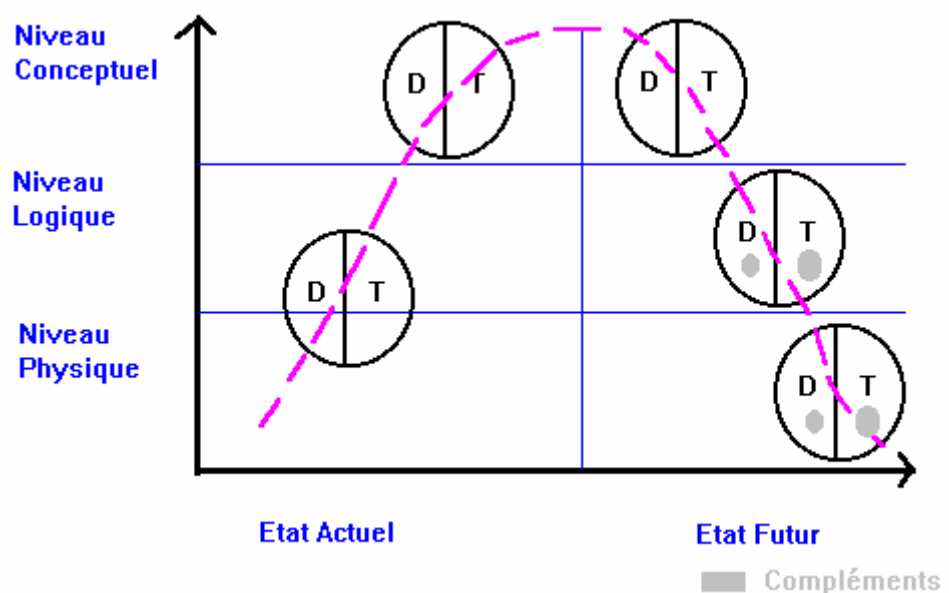
3.D.6 Maintenance

La maintenance des applications va permettre de faire vivre les applications et de les mettre à niveau jusqu'à leur mort.

Elle vise à maintenir le système nouvellement développé en ordre de marche, au moindre coût, à fonctionnalités constantes quels que soient les matériels utilisés ou les évolutions de détails des règles commerciales ou organisationnelles à appliquer.

Pour la maintenance l'influence sur les modèles peut s'appliquer à tous les niveaux, généralement cette influence ne s'applique que sur les niveaux logique et physique. Une modification au niveau conceptuel serait très coûteuse et révélatrice de fautes ou d'oublis graves lors de la conception du système.

La rigueur de conception due à l'utilisation de Merise va en effet permettre de diminuer le coût de maintenance et de prolonger la durée de vie des applications. La maintenance peut agir sur les modèles de la façon suivante :



L'évolution vise quant à elle à maintenir le système nouvellement développé en ordre de marche, au moindre coût, à fonctionnalités croissantes ou par intégration de nouvelles règles commerciales.

Module 4 **Génie logiciel : méthodes et outils**

1. PRÉSENTATION

Cette séquence présente les méthodes et les outils du génie logiciel. Après un énoncé des différents concepts rattachés au génie logiciel, une présentation du marché du génie logiciel est proposée. L'association entre la méthode Merise et l'outil AMC*Designor permettra finalement d'illustrer le rapprochement entre les méthodes et les outils.

1.A. Références techniques

AMC*Designor de la société SYBASE, POWERSOFT comporte un Module de Données, de Communications, de Traitements et un Module Dictionnaire

Designer/2000 et Developer/2000 d'Oracle

Le SGBD Relationnel Oracle pour la gestion du dictionnaire de l'AGL.

1.B. Références bibliographiques

- Le génie logiciel en informatique de gestion - Marcel Soberman
Editions Eyrolles 1992 - ISBN 2-212-08256-8
- Le génie logiciel - Ian Sommerville -
Addison Wesley : ISBN 2-87908-033-9
- Génie Logiciel : Outils de développement d'applications - Collection CXP
Catalogues de progiciels - Edition 1992 Réf H9204 - ISBN 2-86905-265-0
CXP 19-21 rue du Rocher 75008 Paris

2. LES CONCEPTS DU GÉNIE LOGICIEL

Génie : Ensemble des connaissances et des techniques concernant la conception, la mise en oeuvre et les applications de procédés, de dispositifs, de machines propres à un domaine déterminé (génie civil, génie génétique).

Ingénierie : Etude d'un projet industriel sous tous ses aspects à savoir technologiques, économiques, financiers et sociaux (Il s'agit d'associer des personnes de qualification différente -notamment les utilisateurs finaux- pour que l'aspect communication soit partie intégrante du projet). Cette étude nécessite un travail de synthèse coordonnant les travaux de plusieurs équipes de spécialistes qui font autorité dans leur domaine.

Génie logiciel : approche industrielle du développement d'un logiciel à savoir ensemble des méthodes, outils et savoir-faire intervenant dans la production industrielle de logiciels.

Son objectif : l'obtention de produits de qualité dans des conditions optimales de productivité. Il combine les concepts d'ingénierie et de génie.

2.A. l'Atelier de Génie Logiciel (AGL)

Une usine de production industrielle (de voitures par exemple) est souvent organisée en ateliers. Dans chaque atelier le personnel dispose soit d'outils à usage simple ou polyvalent, chacun constitué d'un certain nombre de composants.

L'Atelier de Génie Logiciel ou CASE (Computer Aided System Engineering) comprend un ensemble de composants ou d'outils matériels et logiciels organisés pour la couverture du cycle de production, de qualification et d'évolution de logiciels par des équipes de développement.

Définition d'un AGL dans ses trois dimensions

Atelier	ensemble d'outils cohérent dans un environnement de développement
de Génie	intégrant une démarche méthodologique de la conception à la réalisation et à la maintenance
Logiciel	pour la production d'un logiciel exécutable sur une machine cible, dans un environnement d'exploitation

Les AGL doivent posséder trois fonctions principales:

1. accélérer la fabrication du logiciel grâce à l'utilisation d'outils adaptés à chacune des phases,
2. permettre le travail en parallèle de plusieurs équipes en synchronisant l'ensemble,
3. diminuer les coûts de maintenance ultérieurs en améliorant la qualité de la réalisation.

On distingue trois types d'Atelier de Génie Logiciel selon leur position dans le cycle de développement d'un logiciel :

1. **UPPER-CASE** : produit qui couvre la définition des besoins et la planification ;
2. **MIDDLE-CASE** : couvre la conception, la modélisation, le prototypage, la conception des bases de données ;
3. **LOWER-CASE** : couvre la réalisation du code, les générateurs de code et l'organisation des tests.

2.B. Évolution des Ateliers de Génie Logiciel

Le processus de développement de logiciels met en jeu une infrastructure matérielle et logicielle, dont l'AGL proprement dit ne constitue qu'un sous-ensemble.

Cet ensemble d'équipements, des postes de travail des informaticiens aux ordinateurs centraux avec leurs logiciels, est parfois dénommé plate-forme de développement.

On peut considérer trois générations d'AGL :

2.B.1 La première génération d'AGL

Cette génération est bâtie autour de composants situés sur une machine de développement qui est souvent un terminal qui permet d'accéder à l'AGL qui comporte des bibliothèques de macro-instructions, des langages de troisième et quatrième génération, des éditeurs de texte, des compilateurs et éditeurs de liens, des gestionnaires de bibliothèques.

2.B.2 La seconde génération d'AGL

Avec cette génération d'AGL apparaissent :

- la notion de dictionnaire de données de conception, et de données du SGBD, le tout restant sur un serveur, voire un ordinateur central ;
- les générateurs de code, ce code étant un langage de troisième génération (L3G) comme COBOL, DL/1 ou C, et/ou des macro-instructions formant le squelette des programmes

2.B.3 La troisième génération d'AGL

Cette génération actuelle utilise la souplesse et la puissance des micro-ordinateurs. L'ergonomie du mode graphique est indispensable pour toutes les tâches de conception.

Les postes de travail sont connectés en réseau, la consolidation des informations des projets a lieu autour d'un dictionnaire commun, appelé référentiel, qui peut être sur différents supports, quoique qu'il s'agisse en général d'une base de données relationnelles.

En attendant la quatrième génération !

2.C. Les fonctions et les composants d'un AGL

Voici les six fonctions principales que doit assurer un AGL :

1. La mémorisation des données et des règles dans un dictionnaire de données
2. La modélisation des données et traitements avec au moins deux niveaux d'abstraction : conceptuel et logique
3. L'automatisation du processus de développement avec notamment :
 - la génération des modèles de données logiques et physique,
 - la génération de la documentation
 - la génération du code applicatif
4. La planification du projet et son suivi
5. La gestion de la configuration des applications
6. La qualification des logiciels par le maintien et l'exécution des jeux d'essais

Le processus de développement de logiciels met en jeu une infrastructure matérielle et logicielle, dont l'AGL proprement dit ne constitue qu'un sous-ensemble.

Cet ensemble d'équipements, des postes de travail des informaticiens aux ordinateurs centraux avec leurs logiciels, est parfois dénommé plate-forme de développement.

En général, on distingue les machines cibles sur lesquelles les logiciels développés seront testés, avant passage sur l'environnement de production même si les deux sont parfois confondus.

On peut considérer trois générations d'AGL :

2.C.1 La première génération d'AGL

Cette génération est bâtie autour de composants situés sur une machine de développement qui est souvent un terminal qui permet d'accéder à l'AGL qui comporte des bibliothèques de macro-instructions, des langages de troisième et quatrième génération, des éditeurs de texte, des compilateurs et éditeurs de liens, des gestionnaires de bibliothèques.

2.C.2 La seconde génération d'AGL

Avec la deuxième génération d'AGL apparaissent :

- la notion de dictionnaire de données de conception, et celui du SGBD, le tout restant sur un serveur, voire ordinateur central

- les générateurs de code, ce code étant un langage de troisième génération (COBOL ou DL/1 par exemple), et/ou des macro-instructions formant le squelette de programmes

2.C.3 La troisième génération d'AGL

Cette génération actuelle utilise l'intelligence et la puissance des micro-ordinateurs. Le mode graphique est indispensable pour toutes les tâches de conception, mais aussi par souci d'ergonomie.

Les postes de travail sont connectés en réseau, la consolidation des informations des projets a lieu autour du dictionnaire qui peut être sur différents supports, en général une base de données relationnelles.

En attendant la quatrième génération !

2.D. Les fonctions et les composants d'un AGL

Voici les six fonctions principales que doit assurer un AGL :

1. La mémorisation des données et des règles dans un dictionnaire de données
2. La modélisation des données et traitements avec au moins deux niveaux d'abstraction : conceptuel et logique
3. L'automatisation du processus de développement avec notamment :
 - la génération des modèles de données logique et physique,
 - la génération de la documentation
 - la génération du code applicatif
4. La planification du projet et son suivi
5. La gestion de la configuration des applications
6. La qualification des logiciels par le maintien et l'exécution des jeux d'essais

2.D.1 Les composants fonctionnels

- Le dictionnaire ou référentiel
c'est le support de mémorisation et de contrôle des objets et propriétés des modèles du système d'information manipulé. Ce dictionnaire peut être limité à un usage de développement ou être un outil central d'entreprise (on parle de *Repository*)
- Le modélisateur
ce composant doit faciliter, grâce à un environnement graphique, l'élaboration de tous les modèles, graphes et dossiers représentatifs de la modélisation du système d'information. Il doit également proposer un processus de consolidation des modèles dans le dictionnaire.
- L'éditeur de textes :
pour la gestion des dossiers et des applications.
- L'interface graphique :
la modélisation des concepts manipulés exige une interface graphique.

- Le gestionnaire de projets :
il enregistre les données relatives à la planification, au suivi des travaux et des coûts. Il s'agit souvent d'un produit spécialisé interfacé avec l'AGL.
- Le restructurateur de code (*reverse engineering*) :
ce composant permet à partir des programmes (code source) de rétablir la structure logique des traitements et données.
- Le générateur de programmes, maquettes, prototypes, squelettes de programmes :
ce composant génère un code source en L3G (cobol ou C) à partir d'une description. Certains de ces composants permettent la constitution de maquettes d'écrans, de prototypes ou squelettes des logiciels.
- Le gestionnaire de configuration (bibliothécaire)
Ce composant gère la configuration et la diffusion des niveaux de produits résultants du processus de développement et de maintenance (gestion des différentes versions).
- Le moniteur de tests :
ce composant, associé à des jeux d'essais, permet l'évaluation de la qualité des logiciels
- L'analyseurs de qualimétrie :
ce composant permet le contrôle, la visualisation et l'édition de mesures se rapportant aux critères de qualité (complexité, taille, performance des programmes)

2.D.2 Les composants de services

facilitent le travail des équipes de développement.

- La messagerie :
pour les grands projets, une architecture en réseau est quasiment indispensable pour assurer les fonctions de communications et de partage des informations.
Une messagerie facilite de plus la communication interpersonnelle de notes, circulaires, etc.
- Le métalangage et le métamodèle :
un métalangage associé à un métamodèle, permet de redéfinir la structure du dictionnaire, et donc de l'adapter à l'environnement du projet.
- L'administration des outils et composants de l'AGL
des utilitaires permettent le suivi, le contrôle et l'adaptation de l'outil et des tâches, l'interrogation de la base de données de l'AGL.
Exemple : dans l'AGL Designer 2000 de Oracle, on dispose d'un *Repository Administrator Navigator* et d'un *Repository Administration Utility*.

3. LE MARCHÉ DU GÉNIE LOGICIEL

Dans le domaine du Génie Logiciel, il existe aujourd'hui de très nombreux outils. Depuis les Ateliers de Génie Logiciel (AGL) qui ont pour vocation de couvrir la quasi totalité du cycle de développement jusqu'aux outils les plus spécialisés (outils mono-fonction).

Le CXP (Centre d'eXpérimentation des Progiciels) a édité en 1992 un catalogue intitulé « *Génie Logiciel : Outils de développement d'applications* », il présente plus de 440 outils.

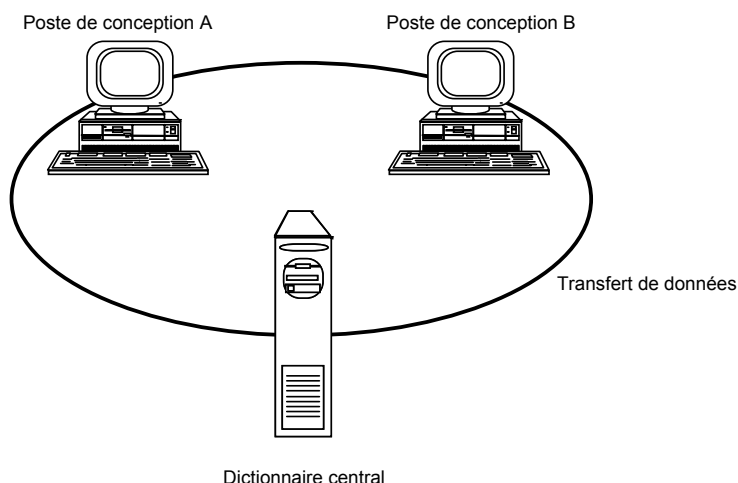
3.A. Les Ateliers de Génie Logiciel (AGL)

Le terme d'AGL n'est ni clair, ni bien défini dans l'esprit des utilisateurs, cela est dû en grande partie aux confusions suscitées par les fournisseurs qui appellent AGL tout outil de développement offrant des aides ponctuelles de Génie Logiciel. En effet, un AGL doit être un ensemble intégré d'outils articulés autour d'un référentiel.

Cette intégration est réalisée par une structure d'accueil ou bus logiciel qui est l'une des composantes essentielles des AGL.

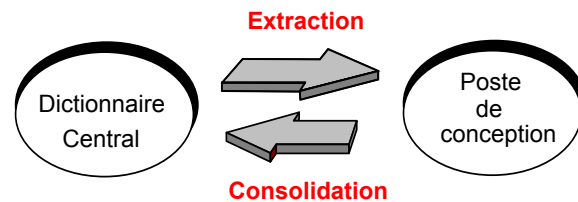
Une structure d'accueil fournit un référentiel de l'ensemble des données utilisées lors du développement de logiciels, ainsi que des mécanismes d'intégration (données et contrôles) et de mécanisme de présentation à l'utilisateur.

Par exemple, avec AMC**Designor* Entreprise un dictionnaire central permet de fédérer les différents des postes de conception



C'est au niveau du dictionnaire central que sont stockées les données de conception. Ce dictionnaire est sollicité par les différents postes de conception au travers d'un module de communication. Il pourra être interrogé directement par un interpréteur SQL.

Les postes de conception sollicitent le dictionnaire au travers d'un module de communication. La circulation des données entre le dictionnaire central et les postes de conception est fondée sur le principe de la consolidation et de l'extraction.



Les postes de conception comportent les modules graphiques de conception de modèles. C'est au niveau d'un de ces postes que pourra être effectuée l'administration du dictionnaire

Vers la quatrième génération d'AGL

Les utilisateurs exigent à présent l'intégration des outils dans un ensemble cohérent, organisé autour de la structure d'accueil et d'un référentiel unique représentant le coeur de l'AGL et permettant aux outils de s'alimenter les uns les autres.

Les deux pistes les plus intéressantes pour l'émergence d'un standard sont :

- AD/Cycle d'IBM et
- PCTE (Portable Common Tool Environnement) de l'Ecma.

3.B. Les Outils de Génie Logiciel (OGL)

Au contraire un Outil de Génie Logiciel (OGL) porte sur une ou deux phases du cycle de vie en non sur toutes les phases.

On distingue, sur le marché des outils de génie logiciel, les familles suivantes:

- les outils de spécification et de conception,
- les générateurs d'applications,
- les générateurs d'interfaces graphiques,
- les gestionnaires de documentation et de reverse engineering,
- les éditeurs,
- les gestionnaires de configurations,
- les outils de test et de qualimétrie.

Parmi les nombreux Outils de Génie Logiciel intervenant à différents phases de développement des logiciels il est fréquent de constater que ces outils ne communiquent pas entre eux.

3.C. Les méthodes utilisées

3.C.1 Intérêt des méthodes

Quels avantages peut-on tirer de l'utilisation d'une méthode ?

Citons-les en vrac :

Tout d'abord, la quasi-totalité des méthodes impliquent des étapes et des points de contrôles obligés, ce qui devra permettre d'avoir une meilleure gestion et un meilleur suivi du projet en cours.

Ensuite, toutes les méthodes, quelles qu'elles soient, impliquent l'utilisation d'un formalisme commun et la constitution d'une documentation, ce qui ne peut que faciliter le dialogue entre les membres du projet d'abord et mais aussi avec les utilisateurs.

La documentation ainsi constituée facilitera à l'évidence la maintenance de l'application, à condition d'éviter toute lourdeur. La maintenance est un poids de plus en plus lourd pour les services informatiques.

En raison de leur approche de la modélisation par niveaux d'abstraction, les méthodes permettent non seulement de concevoir des systèmes sur des bases plus stables, mais aussi de faciliter grandement le dialogue avec les utilisateurs. En effet, contrairement à ce qu'il s'est passé avec les méthodes classiques, ce dialogue sera basé non plus seulement sur l'aspect technique, mais aussi sur l'aspect métier : c'est l'objectif du niveau conceptuel.

Utilisée pour l'ensemble des projets dans le cadre d'une entreprise, une méthode de conception de systèmes d'information comme MERISE permettra d'avoir une vision cohérente de l'ensemble des données de gestion manipulées par les différents domaines de l'organisation.

3.C.2 La pratique des méthodes

Les méthodes vedettes sont :

MERISE en informatique de gestion,

SADT, SART et HOOD en informatique industrielle

L'utilisation de ces méthodes a progressé depuis 1990, au détriment des méthodes internes ou marginales. Cela peut s'expliquer par la plus grande maturité des méthodes "connues", par une meilleure implantation des outils et aussi par le coût important que représente l'entretien d'une méthode propriétaire.

Activité	Informatique Industrielle			Informatique de Gestion		
	Gestion de Projets	Spécification	Conception	Gestion de Projets	Spécification	Conception
Méthode						
Pert	33 %					
Merise		9%	10%	29%	54%	68%
SADT		32%	2%			3%
HOOD			24%			

Source Afcet /Cxp (Enquête sur la pratique du génie logiciel - 1992)

3.D. Les outils utilisés

Il s'agit d'apprécier comment le problème du choix et de l'utilisation des outils informatiques est abordé par les utilisateurs et les développeurs.

Les activités prioritaires en Informatique de Gestion :

Activités	% de citations
Conception	58%
Administration de Données	42%
Documentation	37%
Spécification	35%
Maquettage et Prototypage	31%

Source Afcet /Cxp (Enquête sur la pratique du génie logiciel - 1992)

Les critères de choix d'un outil :

- L'étude d'implantation, l'expérimentation, les démonstrations et les conseils sont dans l'ordre le mode de choix plébiscité par les utilisateurs.
- Efficacité éprouvée et rapport qualité-prix sont les principaux critères de choix des décideurs.

3.E. La méthode Merise et les outils associés

Constatant que l'élaboration manuelle d'un dossier d'étude avec Merise peut s'avérer relativement lourde si le projet dépasse une certaine taille et/ou une certaine complexité, des sociétés de service ont eu la bonne idée de développer des outils d'aide à la mise en œuvre de la méthode.

C'est ainsi que, depuis quelques années, des outils supports à Merise ont fait leur apparition sur le marché, et qu'ils ne cessent d'être améliorés depuis.

Citons entre autres :

- **Principia** de Sema-Metra.
- **PacBase** de la société CGI-Informatique.
- **Adélia** Conception et Réalisation de Hardis International.
- **AMC*Designor** de la société SDP (Modèles de données et de traitements).
Un module développeur, disponible depuis juin 1996 avec la version 5, permet de générer une application dans un contexte d'interface graphique.
- **SECSI** de la société INFOSYS.
- **TRAMIS** de la société Concis.
- **Grafftalk** de la société Parallax.

3.F. MERISE et AMC*Designor

3.F.1 La démarche MERISE pour la modélisation des données avec AMC*Designor

a. Introduction

AMC*Designor facilite la construction des modèles et contribue efficacement à la réalisation de tous les niveaux.

C' est un outil graphique qui utilise le formalisme de la méthode MERISE. Il est composé de deux représentation d'un modèle : conceptuel et physique.

A ce propos, par rapport au cours, avec AMC*Designor le « modèle physique » se situe plutôt au niveau logique et la « génération de base » est en fait le niveau physique.

b. La terminologie de AMC Designor

Le modèle conceptuel :

Au niveau du modèle conceptuel des données (MCD), AMC Designor exploite entièrement les notions d'entité et d'association avec le graphisme préconisé par la méthode.

Une fois le modèle établi, AMC*Designor permet de le vérifier puis de générer automatiquement, par application de règles algorithmiques, un modèle physique des données (MPD).

Le modèle physique :

Ce modèle est la réunion du modèle logique de données (MLD) (relation, attribut) et du MPD (table, colonne). Sa représentation schématique comporte les notions de tables et de colonnes, équivalentes aux notions de relations et d'attributs du MLD, tout en intégrant les notions d'index et d'intégrité référentielle propre aux modèles physiques des SGBD.

Ce MPD peut à tout moment être dénormalisé. L'aboutissement de ces modèles dans la méthode MERISE, comme au niveau d'AMC*Designor, est la mise en place d'une structure de base de données.

AMC*Designor fournit un MPD immédiatement exploitable et se charge de générer le script de création de la base de données.

c. Ressources nécessaires

Pour utiliser AMC*Designor Version 5, vous devez avoir le matériel suivant :

- Processeur PC Intel ou supérieur
- Mémoire vive : 3Mo ou plus
- Windows 3.1x ou version supérieure, Windows NT ou Windows 95
- Espace disque dur : 5Mo par module ou plus

3.F.2 La démarche de conception des modèles de données

ETAPE 1 : le recueil des données

Après une phase de réflexion, on dresse le dictionnaire des données correspondant aux informations relevées sur les documents de gestion ou lors de l'analyse des postes de travail au moyen d'interviews.

AMC*Designor permet la saisie du dictionnaire des données. Il est possible de préciser le nom, le code et le type des données, leur longueur et leur précision. De plus, il assure des contrôles sur l'affectation et la réutilisation des mnémoniques (codes).

ETAPE 2 : modélisation du MCD

AMC*Designor permet de modéliser et d'élaborer le MCD de façon très conviviale à l'aide de l'interface graphique et de la souris. Il permet en outre de contrôler à tout instant la validité et la cohérence formelle et non sémantique de la modélisation.

ETAPE 3 : génération du MPD

Cette opération est effectuée en activant la commande **Générer modèle physique**. Par application des règles de passage préconisées par la méthode MERISE, elle transforme le MCD en un MPD graphique. Il est ensuite possible de travailler directement au niveau du MPD pour optimiser l'implantation de la base ou pour effectuer une dénormalisation (optimisation du schéma de la base).

AMC*Designor demande, lors de l'activation de cette commande, de préciser le choix du système de bases de données (SGBD) cible.

ETAPE 4 : génération de la base de données

Cette étape est effectuée permet, en fonction du SGBD cible choisi, de traduire le MPD en un ensemble de commandes appelé LDD (langage de définition des données). *Voir à ce propos le point de savoir S25 su cours d'architecture logicielle.*

3.F.3 La démarche de conception des modèles de traitements

AMC*Designor facilite la construction du Modèle Conceptuel de Communication, du Modèle Conceptuel des Traitements et du Modèle Organisationnels des Traitements.

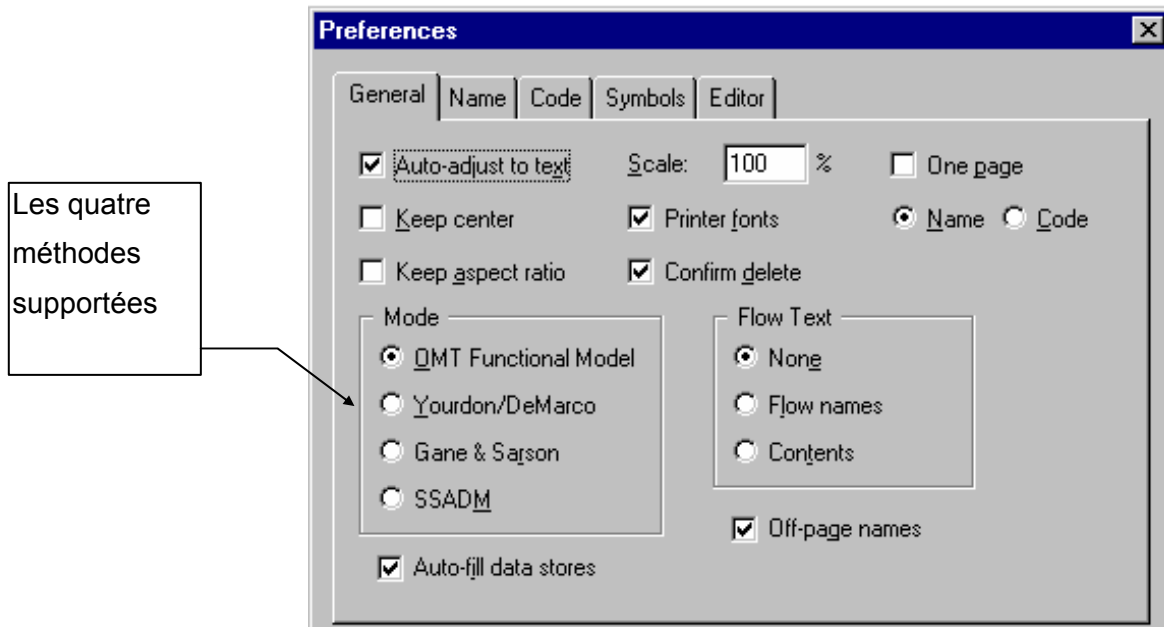
3.F.4 Le module développeur de AMC*Designor

Ce module permet comme le module de données de générer un MPD, puis un script de création de la base de données avec la possibilité de créer des déclencheurs (*triggers*) d'intégrité référentielle.

De plus, il inclut un générateur d'applications pour PowerBuilder ou Visual Basic, qui permet de générer des applications, des fenêtres, des objets.

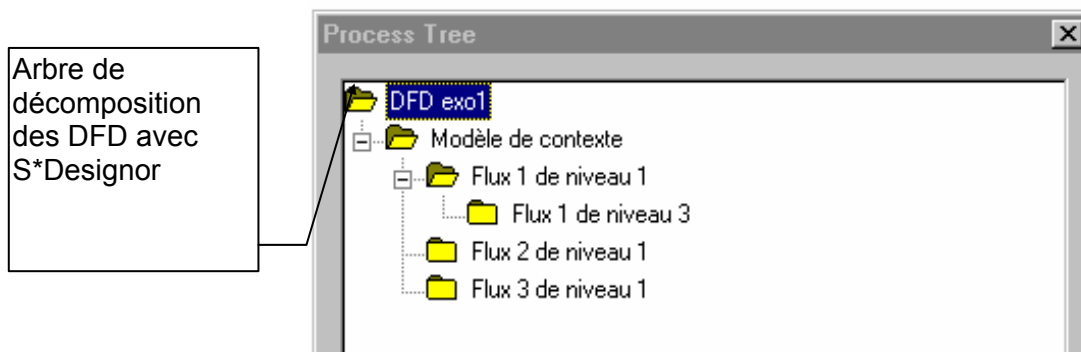
3.F.5 Les Diagrammes de flux de Données avec S*Designer

L'outil S*Designer fournit un module nommé ProcessAnalyst, qui permet d'effectuer la représentation des Diagrammes de Flux de Données (DFD) selon quatre formalismes différents.



Les formalismes supportés correspondent aux méthodes OMT, de Yourdon (méthode d'analyse structurée : SA), de Gane et Sarson et de SSADM utilisée au Royaume Uni.

La décomposition des diagrammes de flux de données est représentée par un arbre hiérarchique :



4. LES ÉTAPES D'UN PROJET INFORMATIQUE

4.A. Découper la méthodologie de développement en étapes

Un projet informatique se découpe en général en deux grandes phases :

La phase de spécifications révèle progressivement les besoins ou l'analyse des besoins du client, de l'utilisateur et de l'informaticien qui ont encore une vue très floue du problème à traiter et de la solution à adopter.

Cette phase se termine par la rédaction des dossiers de spécifications des besoins et de définition du système.

C'est à partir de ces dossiers qu'une décision de réalisation sera prise pour passer à la phase suivante. On rentre alors réellement dans la production du logiciel.

La deuxième phase correspond à la production du logiciel du projet. On distinguera les étapes suivantes du cycle de vie :

- la spécification logicielle
- la conception préliminaire
- la conception détaillée
- la codification et les tests unitaires
- l'intégration et la validation
- la recette
- l'exploitation et la maintenance

4.B. Parcourir les étapes de gestion du projet logiciel

4.B.1 La Phase d'analyse ou de spécification des besoins visés

Cette phase consiste à faire exprimer les besoins précis par les utilisateurs.

4 questions sont posées :

QUESTION	OBJECTIF DE LA QUESTION
pourquoi ?	comprendre le problème (analyse de l'existant, analyse du contexte)
quoi ?	identifier les besoins et produire des spécifications
comment ?	formuler les contraintes de conception et définir la qualité recherchée
quand ?	établir un calendrier

a. Deux activités de l'analyse sont fondamentales

Il faut d'abord résoudre le problème de la communication entre, d'une part, le client et l'analyste et, d'autre part, l'analyste et l'équipe de conception.

L'analyste doit ensuite modéliser les activités, les données, les dialogues, en utilisant des formalismes appropriés, des schémas, etc.

b. Diverses techniques d'analyse peuvent être adoptées

- l'interview du client et des utilisateurs en faisant un effort de communication en utilisant un langage naturel
- la modélisation du problème en utilisant des méthodes (Merise, Sadt,...)
- la construction de prototypes permettant de montrer à l'utilisateur le fonctionnement futur du produit et ainsi de lui permettre de vérifier ce qui a été compris.

4.B.2 Le cycle de vie du logiciel

Il commence dès que la décision de continuer le projet est prise.

Sept activités successives peuvent être réalisées.

a. La spécification du logiciel

Elle dépend entièrement de la phase précédente. Les définitions de la phase précédente vont se préciser jusqu'à devenir des spécifications.

L'analyste a cherché à comprendre le problème et à identifier les besoins en vue d'élaborer une solution.

L'étape de spécifications doit répondre à la question :

« que me demande-t-on de faire ? »

Cette étape aboutit à des spécifications matérielles et logicielles. Ces spécifications sont essentielles puisqu'elles engagent la fabrication du produit.

Elles sont les fondations du développement futur du produit logiciel et servent de base à un accord contractuel avec le client, aux tests de recette, à la planification, à la documentation utilisateur, à l'évolution future du système, à la qualité du produit.

Les spécifications de logiciel aboutissent à un dossier qui décrit :

- les objectifs et fonctions
- l'interface utilisateur (description des dialogues homme-machine)
- les contraintes financières et temporelles
- les conditions d'exploitation et d'utilisation.

b. La conception préliminaire

Il s'agit d'imaginer et de proposer une architecture pour satisfaire les spécifications générales.

Elle consiste en une décomposition en sous-systèmes et établit clairement les frontières entre ces sous-systèmes en se fondant sur la modularité, les procédures d'abstraction des données.

c. La conception détaillée

Approfondit l'oeuvre de conception précédente en détaillant les algorithmes et la structuration des données pour atteindre un niveau satisfaisant qui autorise le codage.

Les idées clefs de la conception détaillée sont : abstraction , élaboration, choix d'alternatives.

Les principales activités sont :

- la décomposition des sous-systèmes en modules ou en tâches,
- la description précise des traitements, des données et des interfaces,
- la résolution d'algorithmes.

d. Le codage et le test unitaire

Ils visent à rédiger et à tester des programmes. Les méthodes de conception conduisent à adopter les principes de la programmation structurée au niveau du codage.

Chaque élément individuel de code doit être testé séparément.

e. L'intégration et la validation

Ils visent à assembler les différents éléments codifiés et à tester la conformité du logiciel par rapport aux besoins.

La recette (ou qualification pour un produit standard normalisé)_fait subir au logiciel une suite de vérifications prédéterminées :

- du respect des spécifications du logiciel et des besoins,
- des performances réelles conformes aux spécifications annoncées,
- de tous les éléments constituant le contexte du logiciel : rapports, manuels, documentation, jeux de test, ...

f. L'exploitation et la maintenance

Elles assurent le suivi logiciel dans sa phase d'utilisation.

Cette étape regroupe les activités suivantes :

- l'utilisation d'un logiciel en exploitation,
- la correction des défauts résiduels (maintenance corrective : 20 %)
- l'adaptation du produit à un nouvel environnement logiciel ou matériel (maintenance adaptative : 20 %)
- la mise à jour du logiciel à la suite de modification des spécifications d'un point de vue fonctionnel ou de performance (maintenance évolutive : 60%)

4.B.3 Les ressources matérielles et logicielles :

Les besoins logiciels vont évoluer tout au long du projet en fonction des tâches à exécuter et des étapes de projet (analyse, développement, tests, validation, formation). Le génie logiciel assisté par ordinateur (CASE) doit permettre aux informaticiens d'obtenir des gains de productivité important.

Pour le développement on utilise aujourd'hui trois types d'Environnements de Développement de Logiciels (SDE : Software Development Environment en anglais) :

1. Les environnements de programmation : ils englobent la programmation, le test et la mise au point,
2. Les AGL : ils sont tournés vers la spécification et la conception du logiciels.

Pour la phase d'analyse l'utilisation des logiciels de Bureautique, des outils de génie logiciel (AMC*Designor et Secsi pour la conception des systèmes d'information), des outils d'aide à la gestion des projets (Ms/Project 3.0, TeamWindows ou des applicatifs spécifiques) permettent d'améliorer la qualité du travail et de diminuer les délais.

Ces outils intègrent depuis peu des interfaces permettant de générer des maquettes fonctionnant avec les outils de développement d'applications Client/Serveur (PowerBuilder, NS-DK, Visual Basic, Sql*Windows ou Centura, Progress pour ne citer que les plus utilisés).

Ils offrent parfois des supports de programmation avec des langages de quatrième génération (L4G).

3. Les environnements de génie logiciel : conçus pour développer de gros systèmes logiciels pour lesquels les coûts de maintenance dépasse les coûts de développement.

Parmi les outils adaptés à la production et à la génération de code pour plusieurs langages informatiques (C, Basic, Cobol, ...) figure la PSP Station qui est basé sur la méthode de Programmation Sans Panne.

Les besoins matériels dépendent des outils logiciels qui déterminent la configuration du poste de travail de l'informaticien. Un micro-ordinateur avec des performances satisfaisantes, relié en général via un réseau à un serveur de base de données devient l'outil de travail le plus fréquent.

4.C. Les modèles de cycle de vie

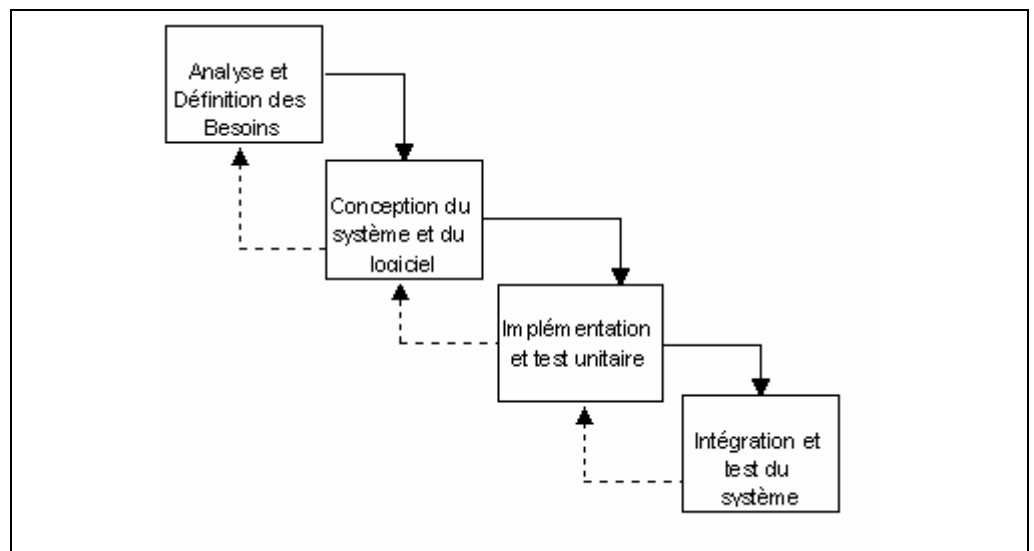
4.C.1 Le modèle en cascade

Nous allons présenter le modèle de processus de conduite de projet esquissé par Boehm dit projet en "cascade".

Les principales caractéristiques de ce modèle sont :

- chaque étape du cycle de vie est caractérisée par des activités dont le but est d'élaborer un ou des produits intermédiaires.
- chaque fin d'étape est matérialisée par un événement (point clé ou pont de rendez-vous) où s'exerce une activité de contrôle (vérification et validation) destinée à éliminer le plus tôt possible les anomalies, les erreurs et les points sombres dans les différents produits réalisés durant cette étape. Le passage à l'étape suivante est conditionné par le résultat du contrôle (acceptation, rejet, ajournement).
- autant que possible les retours en arrière sur les étapes précédentes se limitent à un retour sur l'étape immédiatement antérieure.

Le modèle en cascade



Les flèches en trait plein vers le bas indiquent la validation.

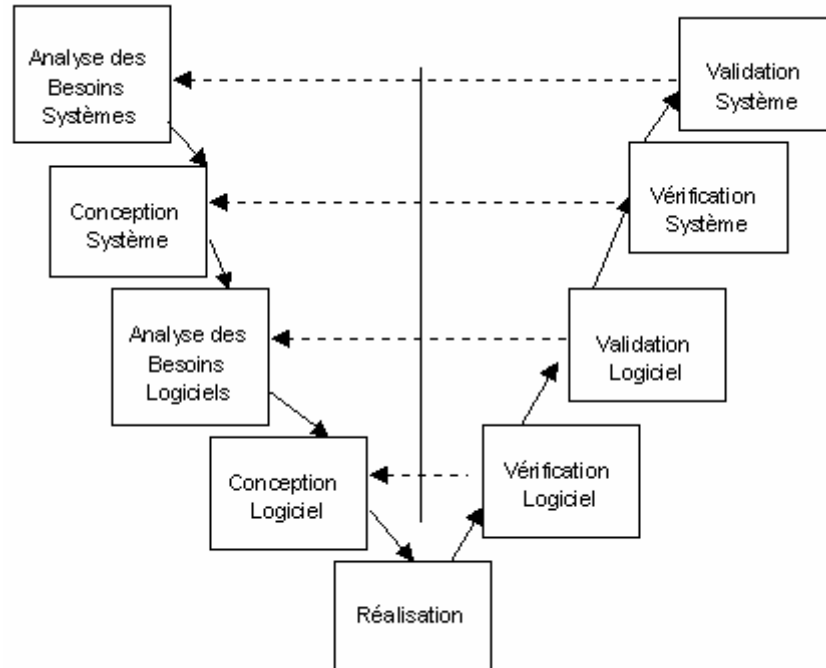
Les flèches en trait pointillé indiquent le retour à l'étape précédente.

4.C.2 Les modèles en V et W

Dans le modèle en V, il s'agit de vérifier que le logiciel est conforme aux spécifications fonctionnelles et techniques, qu'il répond aux besoins du logiciel, que le système est conforme aux spécifications fonctionnelles. On valide la satisfaction des besoins exprimés.

Le terme *système* du schéma ci-dessous doit être pris au sens Système d'Information.

Le modèle en V

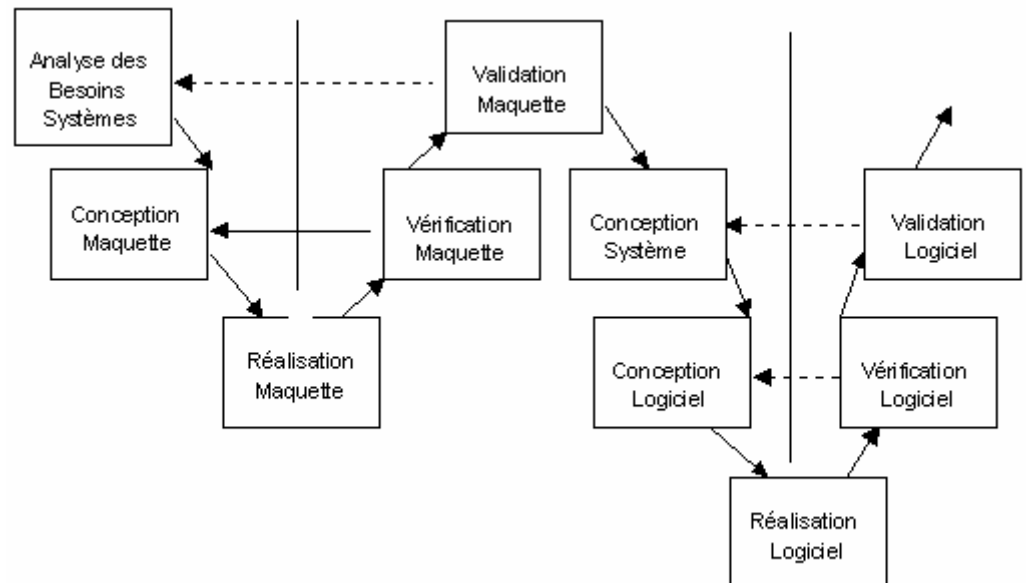


L'inconvénient du modèle ci-dessus est qu'il faut attendre la fin du projet pour avoir le retour des utilisateurs sur le système réellement implémenté.

Pour éviter cet inconvénient, le modèle en V a été aménagé pour permettre des circuits de validation plus courts permettant la mise en oeuvre des techniques de maquetages et de prototypages.

Le modèle intégrant ces techniques a été nommé modèle en W.

Le modèle en W



Un nombre d'itérations (refaire plusieurs fois le même traitement) limité permet de gagner beaucoup de temps sur les étapes en aval et de limiter les mauvaises surprises au moment de la validation finale, alors que le budget du projet est souvent consommé.

On évoque souvent le Développement Rapide d'Applications (RAD : Rapid Application Development en anglais) en parlant de ces modèles.

Module 5 **Modélisation des communications**

1. PRÉSENTATION

Ce module présente la modélisation des flux en s'appuyant premièrement sur la méthode MERISE pour les diagrammes acteurs-flux.

1.A. Références techniques

Les différents modèles qui ont été réalisés dans cette séquence s'appuient sur les produits suivants :

AMC*Designor Module de Communications et de Traitements

S*Designor pour les diagrammes de flux de données

Designor/2000 d'Oracle pour les diagrammes de flux de données.

1.B. Références bibliographiques

- Merise/2 : Modèles et Techniques Merise Avancés - de Panet et Letouche Editions d'Organisations : ISBN 2-7081-1653-3
- Merise par l'exemple - J. Moréjon Les éditions d'organisation. ISBN 2-7081-1290-2
- Pour la **pratique** de la modélisation des données utilisez AMC*Designor 5.1 version évaluation livré sur CD-ROM avec le livre :

AMC*Designor : Mise en œuvre de Merise et Conception d'applications client-serveur par Gilles GUEDJ aux éditions EYROLLES - parution Octobre 96.

2. MODÉLISATION DES COMMUNICATIONS

2.A. Objectifs

La modélisation des communications consiste à analyser la circulation de l'information au sein d'une organisation. Il s'agit de représenter le fonctionnement global du système en représentant les échanges d'informations qu'engendre l'activité de l'organisation étudiée.

L'étude s'applique à un domaine précis qui limite le champ d'investigation de l'analyste et permet de distinguer les communications internes à l'organisation des communications externes qu'elle entretient avec son environnement.

L'objectif de l'étude est la production du Modèle Conceptuel de Communication (MCC). Ce modèle utilise la symbolique du diagramme acteurs/flux.

Le cas « Assistance technique »

Le texte suivant décrit une organisation, nous l'utiliserons pour illustrer cette partie du cours :

« Un éditeur de logiciel propose à ses clients un support téléphonique d'assistance technique. Lorsqu'un client appelle, il obtient un répondeur automatique qui lui propose d'utiliser le clavier de son combiné téléphonique pour solliciter une assistance technique (choix 1) ou une information commerciale (choix 2). Ce répondeur permet au service d'assistance technique de transmettre au service commercial les appels qui lui sont destinés. Après avoir composé le chiffre 1, le client est mis en contact avec un standardiste qui relève son identité et l'objet de son appel. Le standardiste peut refuser l'appel ou l'accepter. Dans ce dernier cas, il transmet l'appel, son identité et son objet à un spécialiste qui écoute le problème et fournit les réponses au client »

Le domaine de l'analyse est ici limité à l'organisation du service de gestion d'assistance technique téléphonique.

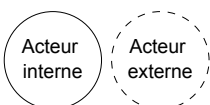
2.B. Vocabulaire

2.B.1 Acteur

Un acteur est un élément émetteur ou récepteur d'objets ou d'information dans le système. Selon le cas, il peut s'agir d'une personne, d'un service ou d'une autre organisation.

On distingue :

- les acteurs internes qui participent directement à l'activité de l'organisation
- les acteurs externes qui sont en rapport avec l'organisation étudiée sans toutefois participer directement à ses activités.



Dans le MCC, un acteur externe est représenté par un cercle tracé en pointillé alors qu'un acteur interne est représenté par un cercle plein. Le nom de l'acteur est placé à l'intérieur du cercle.

Dans le cas « Assistance technique » nous avons souligné les acteurs externes d'un trait pointillé et les acteurs internes d'un trait plein :

« Un éditeur de logiciel propose à ses clients un support téléphonique d'assistance technique. Lorsqu'un client appelle, il obtient un répondeur automatique qui lui propose d'utiliser le clavier de son combiné téléphonique pour solliciter une assistance technique (choix 1) ou une information commerciale (choix 2). Ce répondeur permet au service d'assistance technique de transmettre au service commercial les appels qui lui sont destinés. Après avoir composé le chiffre 1, le client est mis en contact avec un standardiste qui relève son identité et l'objet de son appel. Le standardiste peut refuser l'appel ou l'accepter. Dans ce dernier cas, il transmet l'appel, son identité et son objet à un spécialiste qui écoute le problème et fournit les réponses au client »

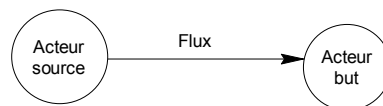
2.B.2 Flux

Flux

Un flux est un transfert d'informations, de biens ou d'argent entre acteurs. Un flux part d'un acteur source pour aboutir à un acteur but, il est matérialisé par une flèche. Un flux est externe si l'acteur source est externe et interne dans le cas contraire.

Un flux d'information est représenté graphiquement par une flèche orientée de l'acteur émettant le flux vers l'acteur le recevant. Le libellé du flux est inscrit à côté de la flèche tracée.

Formalisme graphique



2.B.3 L'ordonnement des flux

Bien souvent les flux peuvent être ordonnés. Il s'agit d'ordonner l'ensemble des flux à partir du graphe acteur/flux (ou de la matrice). L'ordonnement se fera indépendamment de toute périodicité d'émission d'un flux. Un flux annuel pourra par exemple précéder un flux journalier dans l'ordonnement que l'on va produire.

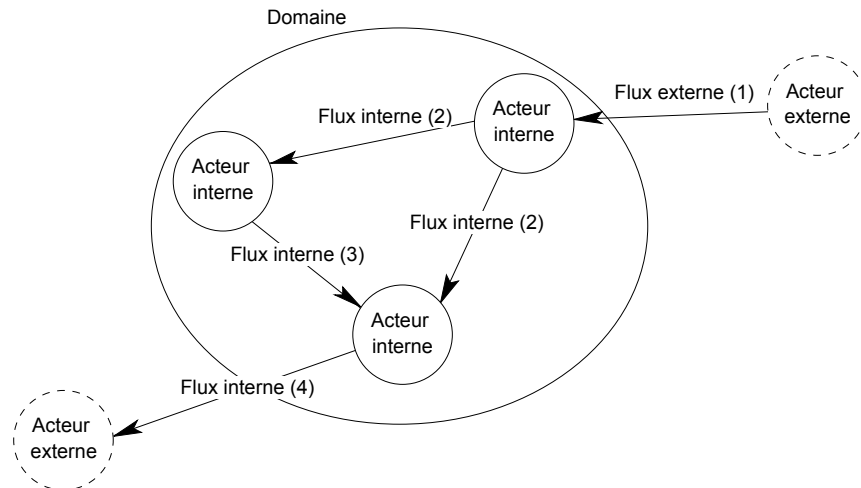
Un **flux primaire** est un flux qui apparaît en premier dans un domaine de gestion.

Exemple : dans un domaine comptable; un flux primaire sera par exemple un bordereau de mouvement, document qui lance toute une chaîne de traitements : journaux, comptes, bilan, etc...

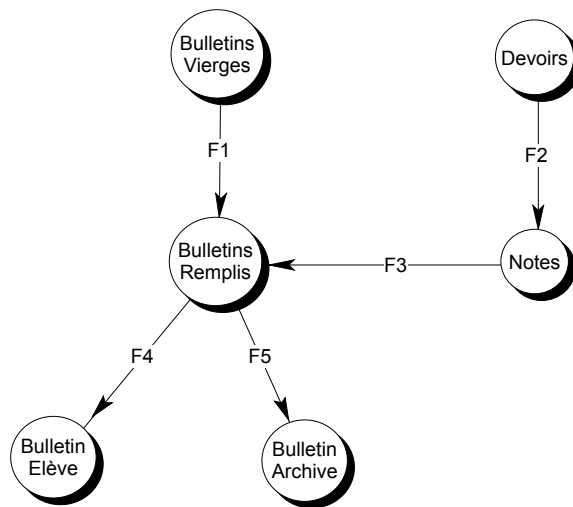
Un **flux secondaire** est un flux dont l'émission est subordonnée à la préexistence d'un ou plusieurs flux primaire ou secondaire.

Ainsi le premier flux (le flux primaire) sollicite une réaction de la part de l'organisation alors que les flux secondaires sont engendrés par cette réaction. Les flux secondaires peuvent avoir lieu dans un ordre déterminé qu'il convient de noter. Cet ordre chronologique n'exclue pas la simultanéité : différents flux peuvent être émis en même temps.

Schéma de principe



Exemple : graphe ordonné des flux concernant les notes d'élève



Dans le cas « Assistance téléphonique » soulignons les flux et ordonnons les :
 « Un éditeur de logiciel propose à ses clients un support téléphonique d'assistance technique.

Lorsqu'un client appelle(1^{er} flux), il obtient un répondeur automatique qui lui propose d'utiliser le clavier de son combiné téléphonique pour solliciter une assistance technique (choix 1) ou une information commerciale (choix 2) (2^{ème} flux).

Ce répondeur permet au service d'assistance technique de transmettre au service commercial les appels qui lui sont destinés. Après avoir composé le chiffre 1, le client est mis en contact avec un standardiste qui relève son identité et l'objet (flux 3) de son appel.

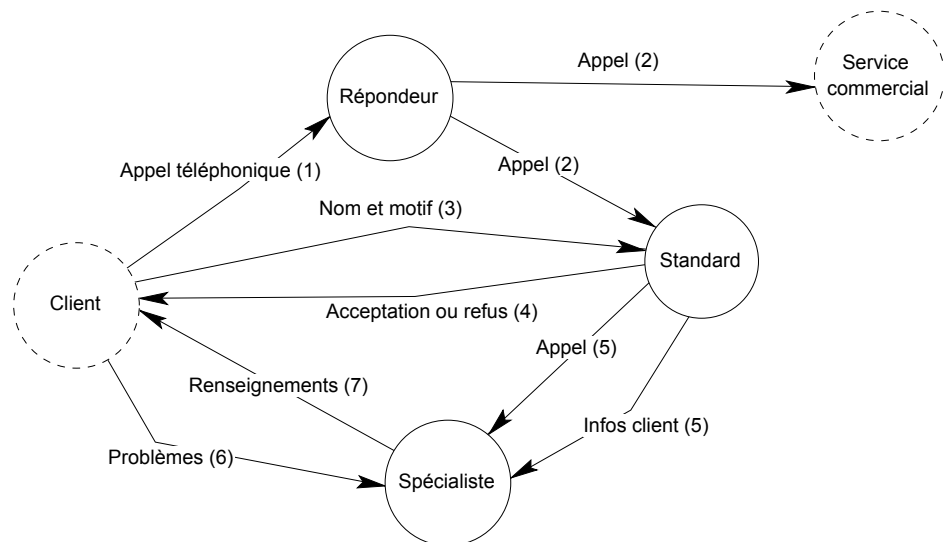
Le standardiste peut refuser l'appel ou l'accepter(ordre 3). Dans ce dernier cas, il transmet l'appel, son identité et son objet à un spécialiste qui écoute le problème et fournit les réponses (ordre 4) au client »

2.C. Méthode

2.C.1 Construction du graphe des flux

1. Identifier le domaine d'étude
2. Repérer les acteurs du système en distinguant les acteurs externes des acteurs internes
3. Repérer les flux entre les acteurs en les ordonnant
4. Tracer le diagramme acteurs - flux

En fait, comme tous les modèles, le MCC sert aussi bien à rechercher comment expliciter le fonctionnement d'un système qu'à représenter ce fonctionnement. Une première ébauche du modèle sert de point d'appui pour affiner l'analyse et trouver d'autres acteurs et d'autres flux. Pour le cas « Assistance téléphonique », on obtient le MCC suivant :



Ce MCC est un modèle à cinq acteurs : deux externes (Client et Service commercial) et trois internes (le Répondeur automatique, le standardiste, le spécialiste). Suivons l'ordonnancement des flux :

Un appel téléphonique reçu par le répondeur constitue un flux externe primaire. Après avoir tapé 1 ou 2, l'appel du client est transmis soit au standard technique soit au service commercial (flux 2). Une fois au standard, le client déclare son identité et donne le motif de son appel (flux 3). Au vu de ces deux éléments le standard annonce au client la prise en charge ou le refus de l'appel (flux 4). En cas d'acceptation (cette condition n'apparaît pas, elle sera cependant présente dans les prochains modèles), l'appel et les informations fournies par le client sont transmises au spécialiste (flux 5). Le client expose son problème au spécialiste (flux 6) qui donne les renseignements attendus (flux 7).

Module 6 **Modélisation des traitements**

1. PRÉSENTATION

Ce module s'appuie sur les caractéristiques générales de la méthode de conception de systèmes d'information : MERISE.

Elle permettra d'aborder la modélisation conceptuelle et organisationnelle des traitements.

Alors que les modèles de données permettent de donner une représentation statique de l'activité d'une entreprise. Les modèles de traitements donnent une vision dynamique de cette activité en explicitant la nature et l'enchaînement des opérations de gestion.

Comme pour les données, l'analyse des traitements est réalisée à deux niveaux :

- **Le niveau conceptuel**

A ce niveau, il s'agit surtout de répondre à la question "Quels traitements et dans quel ordre ?". On cherche à obtenir une représentation schématique des opérations et des événements qui les déclenchent ou qu'elles engendrent. Cette représentation est obtenue avec le MCC (Modèle Conceptuel des Communications) et avec le MCT (Modèle Conceptuel des Traitements).

- **Le niveau organisationnel**

A ce niveau, on répond aux questions "Qui ? Où ? Comment ?". On détaille le MCT en procédures en précisant les délais et les postes de travail concernés. les opérations conceptuelles sont détaillées en procédures élémentaires. Cette représentation est le MOT (Modèle Organisationnel des Traitements).

1.A. Références techniques

AMC*Designor Module de traitements a été utilisé pour réaliser les différents modèles de traitements de cette séquence.

1.B. Références bibliographiques

- Pour la pratique de la modélisation des traitements utilisez AMC*Designor 5.1 version évaluation livré sur CD-ROM avec le livre :AMC*Designor : « Mise en œuvre de Merise et Conception d'applications client-serveur » par Gilles GUEDJ aux éditions EYROLLES - parution Octobre 96.
- Pour approfondir l'étude d'une méthode de conception d'architecture technique utilisez le livre : « Architecture technologique des systèmes d'information : la méthode TACT » de Alaki et Lalanne aux éditions d'Organisation

2. MODELISATION CONCEPTUELLE DES TRAITEMENTS

Le MCT donne une vue globale des traitements réalisés dans le cadre d'un domaine de gestion donné. Il se contente d'identifier les opérations générales (on dira conceptuelles) ainsi que leur enchaînement et les conditions de leur réalisation. Il répond aux questions « Quoi faire ? », « A quelles conditions ? ». Il ne tient pas compte des contraintes organisationnelles (qui ? où ? comment ?).

2.A. Identification des opérations conceptuelles

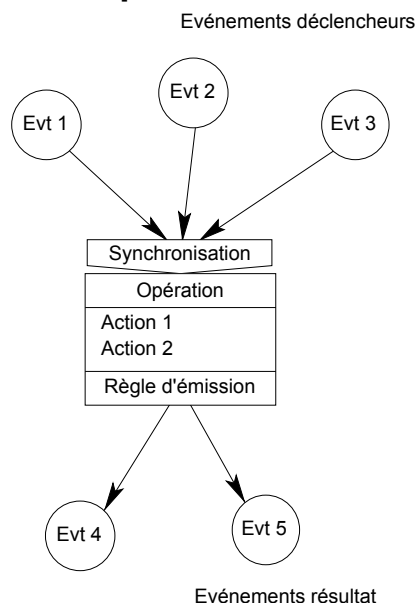
Une opération conceptuelle (OC) représente une ou plusieurs actions de gestion. Son nom doit clairement identifier la ou les actions qu'elle représente.

Une OC est réalisée seulement si un certain nombre d'événements se produisent : on parle d'événements déclencheurs d'une OC. La réalisation de l'opération a également pour effet de créer un certain nombre d'événements : on parle d'événements résultats d'une OC. Un événement résultat peut être déclencheur pour une autre opération conceptuelle.

La règle qui donne précisément la condition de déclenchement d'une opération en fonction des événements déclencheurs est appelée règle de synchronisation. Une règle de synchronisation est une fonction booléenne dont les variables sont les événements déclencheurs. Une telle variable prend la valeur FAUX quand l'événement correspondant n'a pas lieu, elle prend la valeur VRAI quand il a lieu.

Les règles qui donnent précisément la ou les conditions de production des événement résultats sont appelées règles d'émission des résultats.

2.B. Représentation graphique d'une opération conceptuelle



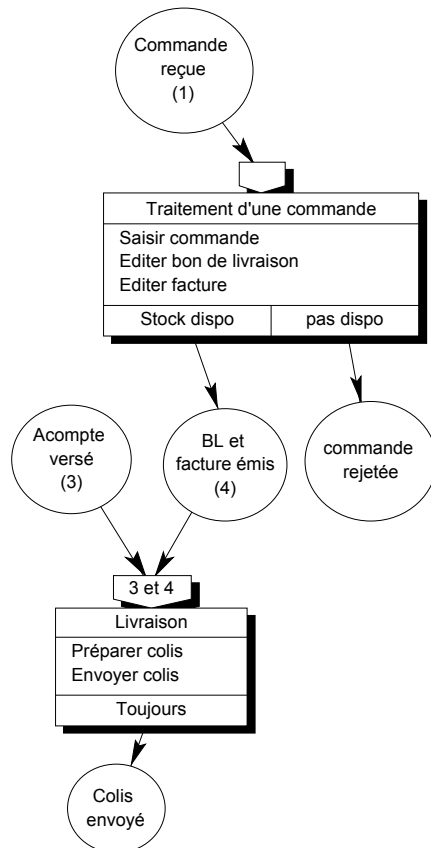
Les événements Evt1, Evt2 et Evt3 participent au déclenchement de l'opération selon une fonction booléenne dite "règle de synchronisation".

L'opération conceptuelle ci-contre réalise les actions 1 et 2.

Elle produit deux événements Evt4 et Evt5 selon une fonction appelée règle d'émission.

2.C. Exemple de MCT

Dans l'exemple suivant, dès qu'une commande est reçue, celle-ci est saisie et, si les produits commandés sont disponibles en stock, la facture et le bon de livraison sont émis. Dans le cas contraire, la commande est rejetée. Une fois la facture et le bon de livraison émis, et dès réception de l'acompte du client, le colis est expédié.



Dans l'exemple ci-contre, l'événement "Commande reçue" est déclencheur pour l'opération "Traitement d'une commande". Cette opération réalise trois actions.

La règle d'émission des résultats est : "Selon que le stock est disponible ou non". L'événement "BL et facture émis" est réalisé si la règle d'émission prend la valeur VRAI, l'événement "Commande rejetée" est réalisé si la règle d'émission prend la valeur FAUX.

L'événement "BL et facture émis" est déclencheur pour l'opération "Livraison" qui réalise deux actions. Cette opération ne se déclenche que quand les événements "Acompte versé" ET "BL et facture émis" sont tous les deux réalisés.

L'événement "Colis expédié" a toujours lieu quand l'opération "Livraison" est exécutée.

2.D. Règles de construction d'un MCT

Pour construire un MCT on pourra partir du modèle conceptuel de communication (voir à ce propos le cours de première année).

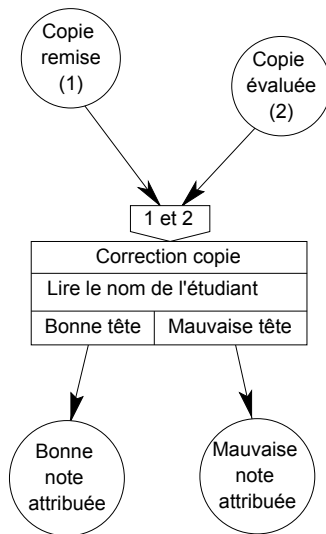
1. Pour chaque flux entre acteurs identifiés dans le modèle de communication on cherchera à lister les opérations qui doivent être menées à bien pour réaliser ce flux. On établit ainsi une liste des opérations qui doivent être réalisées pour aboutir à une gestion cohérente du domaine étudié.
2. Pour chaque opération conceptuelle on établit la liste des événements résultats qu'elle produit ainsi que les règles d'émission correspondantes.
3. On déduit enfin la liste des événements déclencheurs de chaque opération et la règle de synchronisation.

Voici quelques règles qui aident à construire un MCT :

1. Le nom d'un événement est toujours de la forme {nom} {participe passé utilisé comme adjectif}, par exemple : « commande reçue », « délai écoulé », « document imprimé », « cours bien compris ».

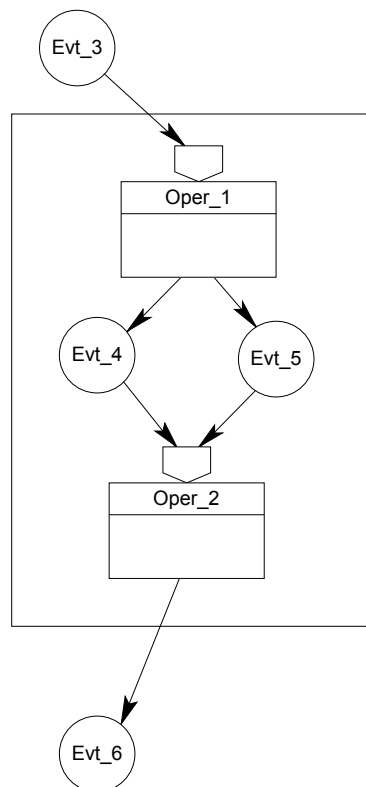
« Commande » n'est pas un nom d'événement correct, alors que « Commande arrivée » est un nom correct.
2. Le nom d'une action est toujours une phrase qui commence par un verbe à l'infinitif.
« saisie des données » n'est pas un nom d'action correct, on préférera « saisir les données ».
3. Une règle de synchronisation est une expression logique construite à l'aide des numéros d'événement et des opérateurs logiques : ET, OU, NON. Elle ne doit pas être toujours fausse.
Par exemple : « Paiement de la traite à échéance si aucun défaut de livraison n'est signalé » n'est pas une règle de synchronisation correcte. On prévoira plutôt les événements :
 - 1) Echéance arrivée
 - 2) Traite reçue
 - 3) Défaut signaléet la règle de synchronisation suivante : « 2 ET 1 ET NON 3 ».
4. Les règles d'émission d'une OC doivent être « complètes » c'est à dire qu'elles doivent représenter tous les cas possibles. Si un cas est prévu, son contraire doit l'être aussi.
La seule règle d'émission « Dossier incomplet » n'est pas suffisante, elle doit être associée à la règle « Dossier complet ».
5. Toutes les actions élémentaires qui dépendent d'une même règle de synchronisation sont regroupées en une seule OC. Autrement dit, l'exécution de toutes les opérations d'une OC n'est interrompue par l'attente d'aucun événement externe.
Une OC qui contiendrait les trois actions suivantes n'est pas correcte.
 - 1) Déposer le permis de construire
 - 2) Consulter les corps de métiers
 - 3) Exécuter les travaux.*En effet, l'action dépend au moins de l'événement externe « Permis de construire accordé ». D'où la nécessité ici de décomposer l'opération conceptuelle en deux opérations distinctes.*
6. Toute opération conceptuelle admet au moins un événement déclencheur et au moins un événement résultat.

7. Tous les événements déclencheurs non résultats sont externes par rapport à l'organisation étudiée (ils émanent d'un acteur externe ou sont fonction du temps).



Dans l'exemple ci-contre, tout est bien entendu correct, sauf l'événement déclencheur n°2. En effet, cet événement est interne (réalisé par un acteur interne), il est donc inutile au niveau conceptuel et peut disparaître.

8. Chaque OC doit admettre au moins un événement déclencheur qui n'est pas résultat d'une autre OC.



Dans l'exemple ci-contre, le déclenchement de l'OC 2 ne dépend que des événements résultats de l'OC 1. On dit qu'il y a « continuité » du traitement entre les deux opérations.

Pour justifier un tel découpage, l'OC 2 devrait nécessiter la réalisation d'un événement externe.

Sinon, les deux OC peuvent être agrégées en une seule OC.

9. Tous les résultats finaux produits par l'organisation doivent apparaître en tant qu'événements résultat.

2.E. Validation

Une fois la première version d'un MCT dessinée, il convient de procéder à une phase de validation en vérifiant point par point les règles ci-dessus. On procédera, le cas échéant, à une agrégation d'OC ou, au contraire, à une décomposition.

Comme tous les modèles d'analyse, la création d'un MCT est une opération itérative constituée successivement d'opérations de création, de validations et de modifications.

2.E.1 Cas d'une agrégation

Soit OC1 et OC2, deux opérations conceptuelles, telles que tous les événements déclencheurs de OC2 sont des événements résultats de OC1. Dans ce cas, OC1 et OC2 peuvent être agrégées en une seule et même opération conceptuelle.

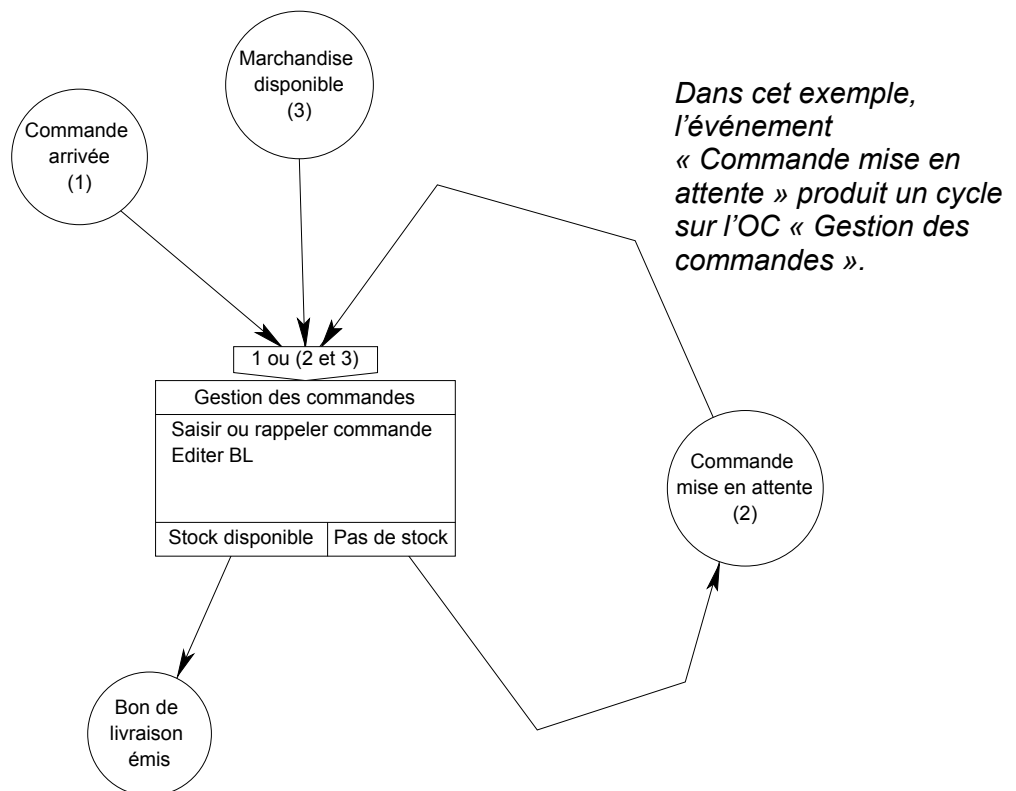
2.E.2 Cas d'une décomposition

Soit OC1 une opération conceptuelle, si la réalisation d'une ou de plusieurs actions internes nécessite la réalisation d'événements externes non déclencheurs de OC1, il faut scinder l'opération en différentes OC.

2.F. Cycles

Un événement résultat pour une OC peut être événement déclencheur pour la même OC ou pour une OC antérieure.

Par exemple :



2.G. Délais

La représentation des délais d'exécution des opérations est importante dans les MCT. Il convient de savoir quand et comment les modéliser.

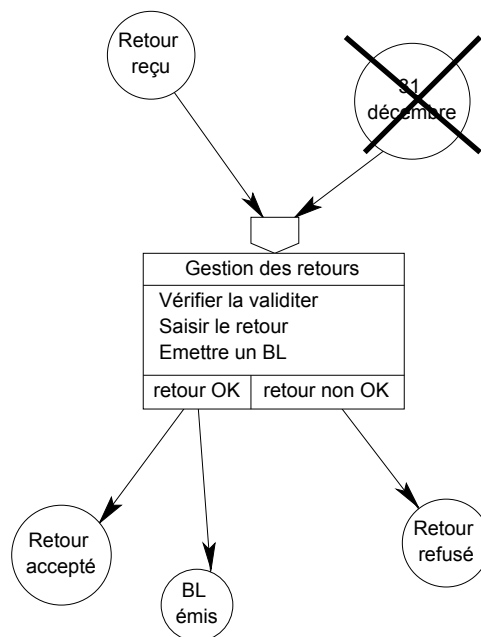
Pour être représenté comme événement dans une OC, l'arrivée à terme d'un délai doit souscrire à deux conditions :

- l'arrivée du terme participe directement au déclenchement de l'opération

- l'arrivée du terme est de nature conceptuelle, c'est à dire externe au domaine et pouvant provoquer une réaction de la part des acteurs du domaine.

Exemple 1

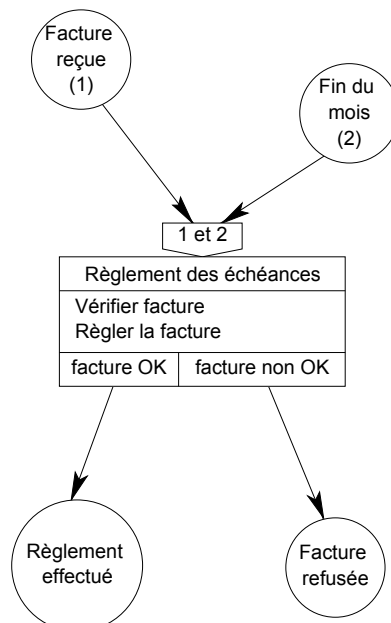
Suite à un défaut de fabrication, une société accepte d'opérer un échange standard sur un produit précis s'il est retourné avant le 31 décembre de l'année en cours.



Ici, le délai « 31 décembre » ne déclenche pas d'action par lui-même. Il est seulement utilisé en interne pour traiter l'action « Vérifier la validité ».

Exemple 2

Le service comptable d'une société règle certaines échéances en fin de mois après avoir vérifié la validité des factures.



Ici, le délai « fin du mois » déclenche l'action de vérification des factures.

Il est donc pertinent de le faire apparaître comme événement déclencheur.

3. MODELISATION ORGANISATIONNELLE DES TRAITEMENTS

Au niveau organisationnel, l'analyse des traitements répond aux questions « QUI ? OU ? QUAND ? ». L'analyse détaille les résultats obtenus à l'étape conceptuelle.

Le MOT (Modèle Organisationnel des Traitements) est un schéma qui donne une vue détaillée des traitements réalisés dans le cadre d'un système d'information. Il intègre les notions de délais (quand ?) , de ressources (avec quoi ?) de lieu (où ?) et de responsabilité (avec qui ?).

3.A. Les procédures fonctionnelles

Chaque opération conceptuelle (OC) du MCT se décompose en procédures fonctionnelles (PF) ou tâches. Chaque procédure fonctionnelle représente la réalisation d'une action élémentaire et ininterrompible par un poste de travail du système.

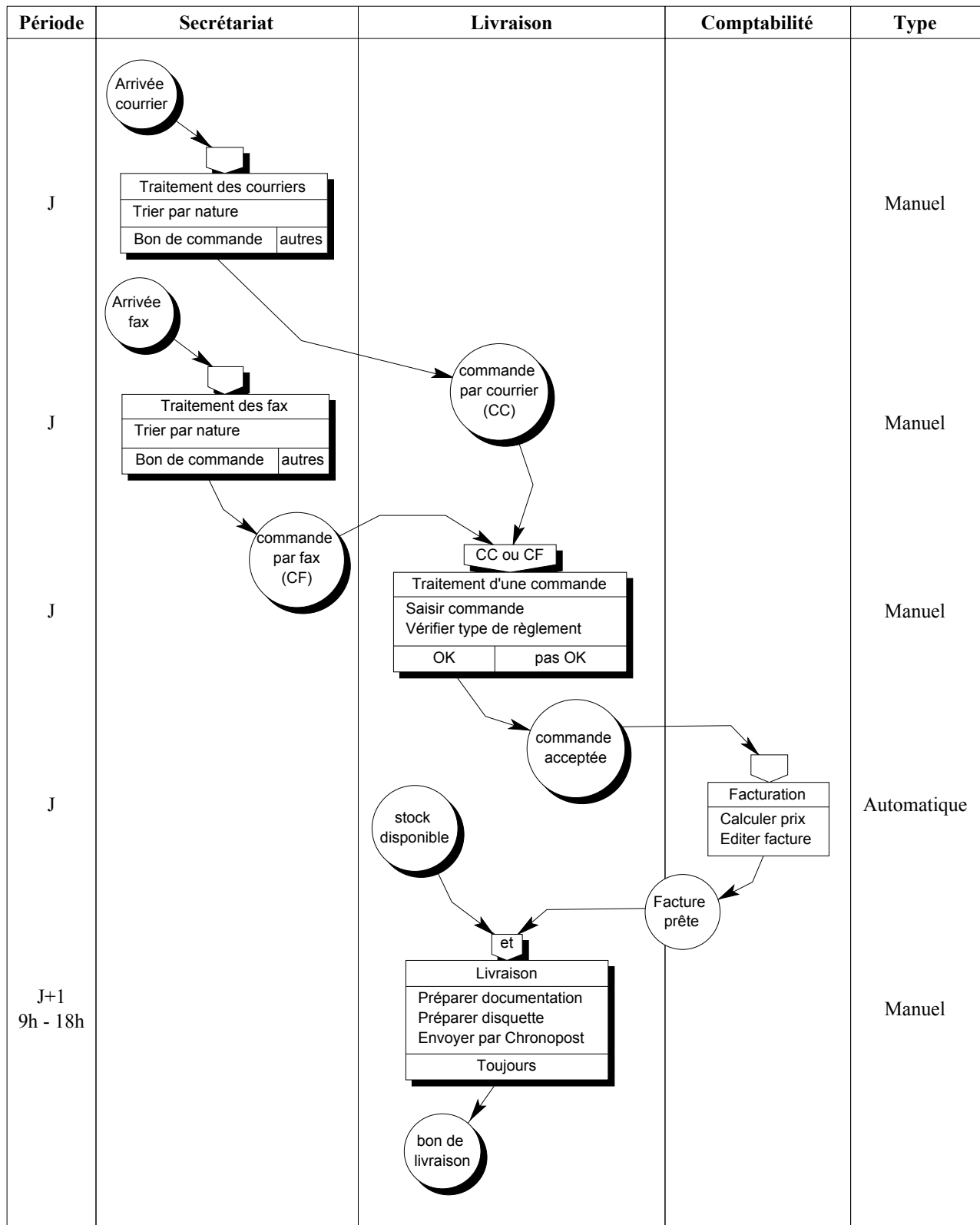
Pour chacune d'elle on précisera :

- le début et la durée de la procédure
- les événements et flux entrant
- les événements et flux sortant
- les règles d'émissions des flux et événements sortant
- la règle de synchronisation des flux et événements entrant
- le type de traitements réalisé : M (manuel), TR (temps réel, dialogue direct à l'écran), TD (temps différé, exécution sans intervention de l'utilisateur)
- la description du poste de travail concerné : nom du responsable, lieu, ressource en personnel et en matériel utilisées.
- après validation avec le MCD, on ajoutera la ou les sources de données (tables d'un SGBD ou fichiers) concernées ainsi que les modes d'accès (lecture, écriture, modification).

3.A.1 Représentation du MOT

Les délais de réalisation des tâches, les postes de travail ainsi que les types de traitements sont représentés en colonnes. On pourra éventuellement substituer les colonnes poste par une seule colonne qui contient le nom du poste qui prend en charge une opération.

Le modèle ci-après est fourni, à titre d'exemple, avec l'AGL AMC*Designor.



3.A.2 Les événements organisationnels

Les événements et flux entrant et sortant d'une tâche illustrent le transit de l'information au sein du système d'information. Les flux entrant représentent l'ensemble des informations qui seront traitées lors d'une exécution de la tâche.

Les événements et flux sortant représentent l'ensemble des informations émises lors d'une exécution de la tâche. Un flux sortant est soit un résultat pour l'univers extérieur, soit un flux entrant pour une autre tâche.

Les flux sont véhiculés sur des supports (voix, papier, écran, support magnétiques, électronique, etc.).

3.B. Validation Données - Traitements

Jusqu'à présent l'analyse des données d'une part et celle des traitements d'autre part ont été menées parallèlement dans le temps sans qu'aucune concordance ne soit recherchée. L'étape de validation a pour objectif de tester la cohérence du modèle des données par rapport au modèle des traitements. Autrement dit, il s'agit de répondre aux questions suivantes : les traitements envisagés sont-ils possibles avec les données modélisées? Les données existent-elles, les accès à ces données sont-ils prévus ?

3.B.1 Procédure de validation Données / Traitements

On pourra procéder de la façon suivante :

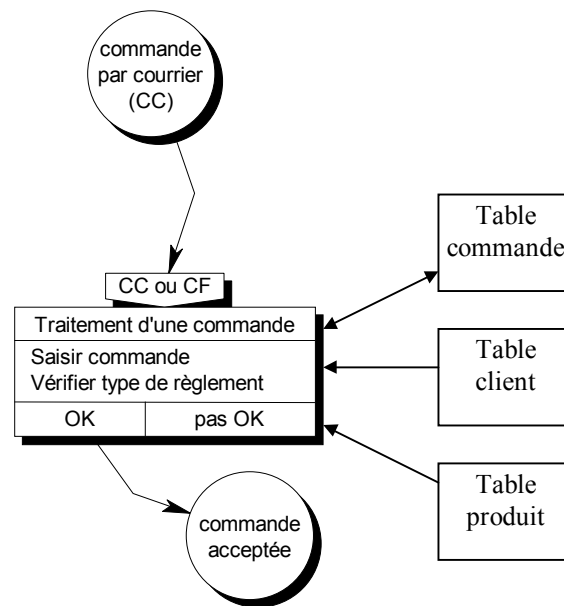
1. Pour chaque tâche du MOT qui décrit un traitement automatisé, on trace un *modèle externe* des données (ou vue externe). Un modèle externe est un "petit" MCD obtenu uniquement à partir des données utilisées dans la tâche et en fonction des règles de gestion de cette même tâche.
2. On compare l'ensemble des données du modèle externe avec le dictionnaire des données. Le premier doit être un sous ensemble du deuxième.
3. On compare les moyens d'accès (clé, associations, cardinalités...) du modèle externe et ceux du MCD. Le premier doit être un sous ensemble du deuxième.
4. Si l'appartenance du modèle externe au MCD n'est pas prouvée, ce dernier doit être adapté en conséquence.

Une fois la validation réalisée on peut ajouter une colonne au MOT afin de donner la liste des tables ou fichiers concernés par chaque PF du MOT.

3.B.2 Représentation des tables et/ou fichiers

Lorsqu'une tâche est automatisée, elle peut utiliser des données stockées dans des fichiers (séquentiels, relatifs, indexés) ou dans des tables sur un SGBDR. Si tel est le cas, on ajoute en correspondance de la procédure concernée, des flèches et des carrés qui matérialisent l'utilisation des données : création (flèche à droite), consultation (flèche à gauche), mise à jour (double flèche), suppression (flèche à droite barrée).

Par exemple :



Dans le cadre d'un Modèle Opérationnel des Traitements (MopT), on pourra, pour chaque tâche correspondant à un traitement automatisé, rédiger le ou les algorithmes ou requêtes qui décrivent l'exécution de la procédure.

4. ARCHITECTURE TECHNIQUE

4.A. Définition

Les entreprises ont aujourd'hui beaucoup de mal à maîtriser, organiser et rationaliser leurs investissements informatiques et de communication.

Cela est dû à la grande rapidité des évolutions technologiques, à la richesse grandissante de l'offre technique, au dynamisme commercial des constructeurs et des éditeurs, mais aussi aux besoins toujours plus exigeants des utilisateurs et des informaticiens.

Définition

L'ARCHITECTURE TECHNIQUE concrétise matériellement les objectifs stratégiques et la conception du système d'information d'une organisation . Elle est défini à deux niveaux

- Un niveau logique qui est stable, réactualisable et ne dépend que de l'organisation choisie. Il débouche sur une **architecture logique** décrivant des scénarios de répartition des données et des traitements, ainsi qu'une description technique qualitative et volumétrique de ses différents composants.
- Un niveau physique qui propose des solutions techniques s'appuyant sur l'architecture logique retenue et débouche sur une **architecture physique**. Elle décrit les équipements, le réseau et les logiciels nécessaires, conditionne le budget et les plans d'action et peut conduire à la rédaction des cahiers des charges. Il est fortement conditionné par l'offre technologique du moment et les contraintes qu'implique un état informatique existant. Il décrit les dates et actions clés de la trajectoire visant à mettre en œuvre l'architecture technique retenue.

Les composants de l'architecture technique

Les composants de l'architecture et leur évolution technologique (tellement rapide en informatique) sont à la fois l'une des causes et la conséquence des enjeux précédents. Ces composants peuvent être présenter selon 3 axes majeurs :

Axe 1 : gestion de l'information

C'est l'axe de l'informatique traditionnelle, matériel et logiciel. On y retrouve tous les éléments et outils qui traitent, manipulent, stockent et restituent l'information.

Axe 2 : gestion de communication

Cet axe rassemble l'ensemble des organes, des procédures et des modes de communication qui constituent le squelette de toute architecture technique. Il intègre aussi les outils d'exploitation du réseau.

Axe 3: normalisation

L'axe de la normalisation doit orienter le choix des solutions physiques pour assurer leur portabilité dans des environnements hétérogènes, leur transparence et leur sécurité d'évolution.

Pour le détail des différents composants matériels et logiciels utilisés dans les architectures techniques récentes vous pourrez vous référer aux séquences de cours présentant tous ces composants (Cours S1 pour le matériel et cours S2 pour les logiciels)

Les modèles d'architecture technique

Les trois axes présentés brièvement précédemment conduisent à des modèles d'architecture technique type qui seront **personnalisés** par **le contexte et le cadre** de l'étude entreprise.

La dimension géographique impose l'optimisation des graphes de connexion. Les besoins de répartition des données et des traitements conduisent à envisager plusieurs modèles d'architecture, qui impliquent généralement une étude soignée des opportunités offertes par les nouvelles technologies (SGBD Relationnels répartis, transactionnels coopératifs, répartition des ressources physiques....).

La difficulté de conception d'une architecture technique consiste donc à gérer le compromis entre :

1. l'axe de gestion des informations et les fonctionnalités requises par l'utilisateur,
2. l'axe de gestion des communications et la nécessité d'échanger d'autant plus que l'informatique est répartie,
3. l'axe des normalisations pour garantir l'ouverture et la pérennité de l'architecture, le coût global de la modernisation et les évolutions technologiques.

Les modèles d'architecture sont donc la résultante d'autant d'impératifs contradictoires. Pour pouvoir dégager ces modèles, il faut s'enraciner dans l'ensemble des études conceptuelles et organisationnelles déjà entreprises, s'adapter au cadre et au contexte et rassembler l'ensemble des compétences nécessaires.

C'est ainsi qu'apparaissent de nouveaux métiers indispensables à l'évolution technique et fonctionnelle de l'informatique :

- le concepteur et le gestionnaire de l'architecture technique,
- **l'administrateur et le gestionnaire du réseau** dont le rôle est d'assurer l'évolution du réseau, sa qualité de service et son succès auprès des utilisateurs,
- l'administrateur et le gestionnaire des bases de données,
- le formateur.

4.B. Conception d'une architecture technique avec TACT

4.B.1 Une méthode

C'est en effet, l'analyse des choix stratégiques et organisationnels de l'entreprise qui permet de construire, à l'aide d'une modélisation, une architecture technique dont la finesse dépend du stade de l'étude.

Toute opération de modélisation fait référence aux niveaux d'abstraction : conceptuel, logique, physique déjà pratiqués par la mise en œuvre de la méthode Merise.

TACT est une méthode de conception d'architecture technique qui a été définie en complément à la méthode MERISE. Cette méthode peut être positionnée par rapport aux niveaux d'abstraction proposés par la méthode Merise de la manière suivante :

Niveaux d'abstraction	Modèles de Données	Modèle de Traitements	Moyens informatiques
Conceptuel	Vues Externes (VED) Modèles Externes (MED) Modèle Conceptuel (MCD)	Modèle Conceptuel (MCT) Graphe des Calculs (GC)	Interprétation et modélisation
Logique ou Organisationnel	Modèles Externes de Données (MED) Modèle Logique (MLD)	Matrice des Flux de Données (MFD) Diagramme des Flux de Données (DFD) Graphe de dépendance des Documents (GDD) Diagramme de Circulation des Documents (DCD) Modèle Organisationnel (MOT)	Architecture logique
Physique ou Opérationnel	Modèle Physique (MPD)	Modèle Opérationnel (MOPT)	Architecture physique
MERISE			TACT

La démarche de conception d'une architecture technique associe à chaque niveau d'abstraction des objectifs et des modalités précis :

1. le niveau conceptuel : vise à interpréter les cibles fonctionnelle et organisationnelle et à définir des unités d'œuvres techniques qui seront des invariants pour les niveaux suivants ;
2. le niveau logique : vise à concevoir les scénarios d'architecture qui répondent au mieux aux spécifications fonctionnelle et organisationnelle ainsi interprétées et aux objectifs et contraintes de l'entreprise. L'Architecture Logique définit les modèles de répartition des données et de traitements, les fonctionnalités techniques requises, la description qualitative et volumétrique des différents composants ;
3. le niveau physique : cherche à répondre aux spécifications de l'Architecture Logique par les solutions physiques les mieux adaptées au contexte de l'entreprise.

La méthode TACT propose une démarche rigoureuse et adaptée aux différentes étapes de conception d'une architecture technique :

1. Schéma Directeur (orientations et esquisses de scénarios).
2. Etude préalable (opportunités et validation des choix).
3. Etude détaillée (étude technique fine).

Les choix techniques et les investissements qui en sont les conséquences sont d'une importance capitale car ils engagent en effet, l'entreprise à moyen et long terme et conditionnent souvent ses capacités de réponse face à des enjeux majeurs :

- objectifs de positionnement dans un environnement concurrentiel,
- enjeux de productivité,
- de qualité et de niveau de service,
- objectifs commerciaux ou technologiques,
- maîtrise de l'information de production ou de pilotage,
- motivation et dynamisme des équipes informatiques au service des utilisateurs.

La conception d'architecture technique en informatique réalisée avec méthode et rigueur, s'impose comme le seul moyen de traduire en termes techniques les réflexions conceptuelles ou organisationnelles d'un schéma directeur ou d'une étude préalable.

4.B.2 Dans un environnement complexe

La conception d'une architecture technique informatique n'est pas une opération isolée de tout contexte. Elle s'insère en effet dans un environnement complexe, sous l'influence de plusieurs facteurs et conjonctures.

L'entreprise définit des orientations stratégiques qui tiennent compte de ses forces et de ses faiblesses, des opportunités et risques du secteur d'activité, d'un système de valeur et d'une culture d'entreprise. Lorsqu'elle fait appel aux technologies de l'information, l'entreprise a déjà un certain nombre d'enjeux.

L'architecture technique, les moyens mis en œuvre et les systèmes logiciels de base prévus doivent également s'adapter à une évolution technologique qui s'accroît d'année en année. L'effort prospectif et imaginatif est devenu un des préalables principaux à la conduite d'une étude d'architecture technique.

Les équilibres souvent fragiles ne doivent pas être bouleversés et c'est ici l'adéquation de la méthode aux contextes d'intervention qui est prioritaire.

Un schéma directeur, des études préalables ou l'analyse d'un système existant conduisent à appliquer différemment la méthode de conception proposée, au cours du cycle de vie de l'architecture technique.

Cet ensemble de contraintes d'environnement, conduit à considérer que toute méthode de conception d'architecture technique doit rendre cette architecture

- portable,
- modulaire,
- cohérente,
- évolutive,
- économique,
- sûre et
- exploitable.

4.C. Exemple d'architecture technique

Pour illustrer notre propos nous avons exploité le contenu d'un cahier de charges décrivant l'architecture technique de la Communauté Urbaine de Strasbourg.

a. Le contexte

La Communauté Urbaine de STRASBOURG a été créée en 1968. Avec ses 6.000 agents, 60 services et ses 200 lieux de travail différents, elle recouvre un nombre important de métiers, de savoir-faire et d'expériences.

En 1990, la Communauté Urbaine de STRASBOURG a décidé de moderniser son administration. Dans le cadre de ce processus de modernisation, la Communauté Urbaine de STRASBOURG a lancé un Schéma Directeur Informatique.

De ce Schéma Directeur, ont découlé plusieurs études qui ont conduit en particulier à la réalisation des projets de câblage (courant 1993), à l'équipement des services en outils micro informatiques (projet Bureautique de 1993 à 1995), à la mise en place d'un réseau informatique (1994 à 1995) et la généralisation d'une messagerie micro informatique (MS-MAIL en 1995).

Une réactualisation du Schéma Directeur des Systèmes d'Information et de Télécommunications débutera en janvier 1997.

b. Le parc micro-informatique (pour information)

Le parc micro-informatique de la CUS compte actuellement 1 500 postes (1 400 P.C. compatibles et 100 Macintosh) dont plus de 1 200 connectés en réseau local Netware 3.12 ainsi qu'environ 600 imprimantes de type HP 4 dont plus de 200 sont également gérées par les serveurs Netware. Les P.C. sont en Windows 3.1 et les Macintosh en système 7.

c. Description de l'Architecture du réseau mis en place à la CUS

Dans le cadre du programme de déploiement d'une nouvelle structure informatique, la Communauté Urbaine de Strasbourg (CUS) s'est dotée d'un câblage informatique structuré et a mis en place un réseau qui compte aujourd'hui 1 000 postes connectés.

Les systèmes sont de 3 niveaux :

- **personnel** (DOS/WINDOWS ou MAC),
- **départemental** (serveurs applicatifs Unix, AS400, etc.),
- **central** (serveur MVS).

Ces différents systèmes sont reliés au moyen d'un réseau fédérateur Ethernet (802.3). Le câblage est de type structuré, à base de paires torsadées (câble et connectique d'Infra+). Ce câblage permet le raccordement des équipements selon le protocole 10 Base T (10BT).

Les postes de travail individuels (P.C., Unix et MAC) sont reliés sur un réseau Ethernet, dit départemental, dont la couverture géographique correspond usuellement à la zone d'influence des locaux techniques du câblage. Ces réseaux départementaux fédèrent environ 80 à 130 postes de travail

individuels. Chacun d'entre eux est piloté par un serveur Bureautique fonctionnant sous NETWARE 3.12. Ainsi, le partage des ressources Bureautique est transparent pour les P.C., les postes de travail sous Unix et les MAC. Chaque réseau départemental est doté d'un serveur Bureautique et de quelques périphériques (imprimantes). 10 réseaux de ce type sont mis en œuvre au centre administratif.

Chaque service est rattaché à un serveur Bureautique, la majorité des postes de travail d'un service étant regroupés sur un même réseau départemental.

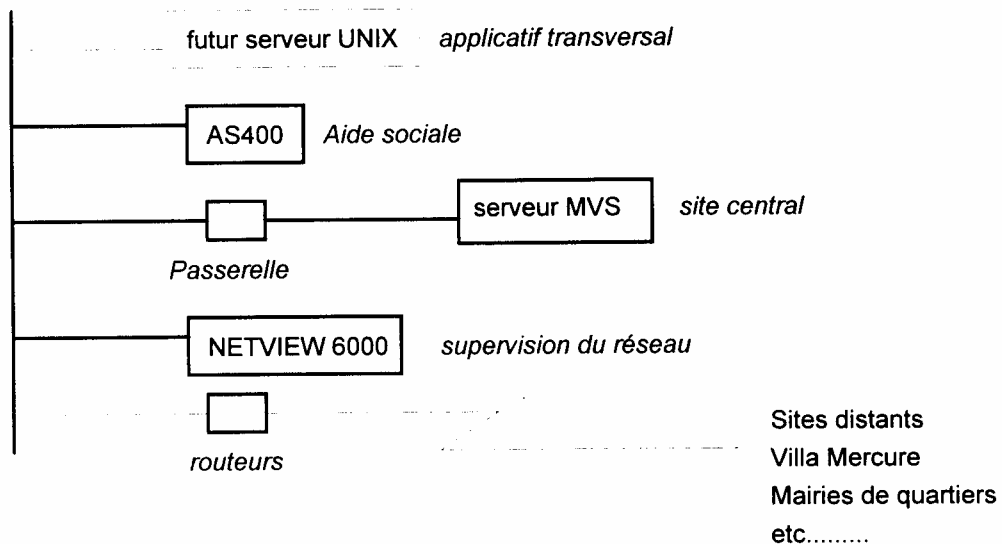
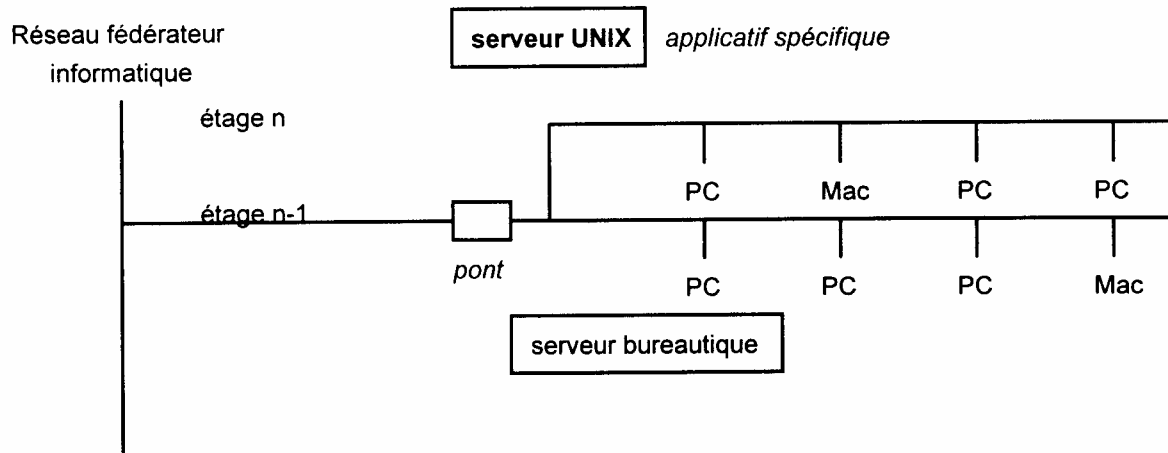
Outre les serveurs Bureautique, les réseaux départementaux sont également desservis par des serveurs applicatifs UNIX. Les postes de travail (P.C. ou MAC) ont accès aux serveurs applicatifs sous Unix et au serveur MVS, indépendamment de leur localisation, grâce au logiciel PATHWAY ACCESS de WOLONGONG.

La séparation logique entre les flux du réseau fédérateur et des sous-réseaux départements se fait au moyen de ponts locaux 802.3/802.3. Des serveurs applicatifs « transversaux » (i.e. commun à plusieurs services) sont également connectés sur le réseau fédérateur.

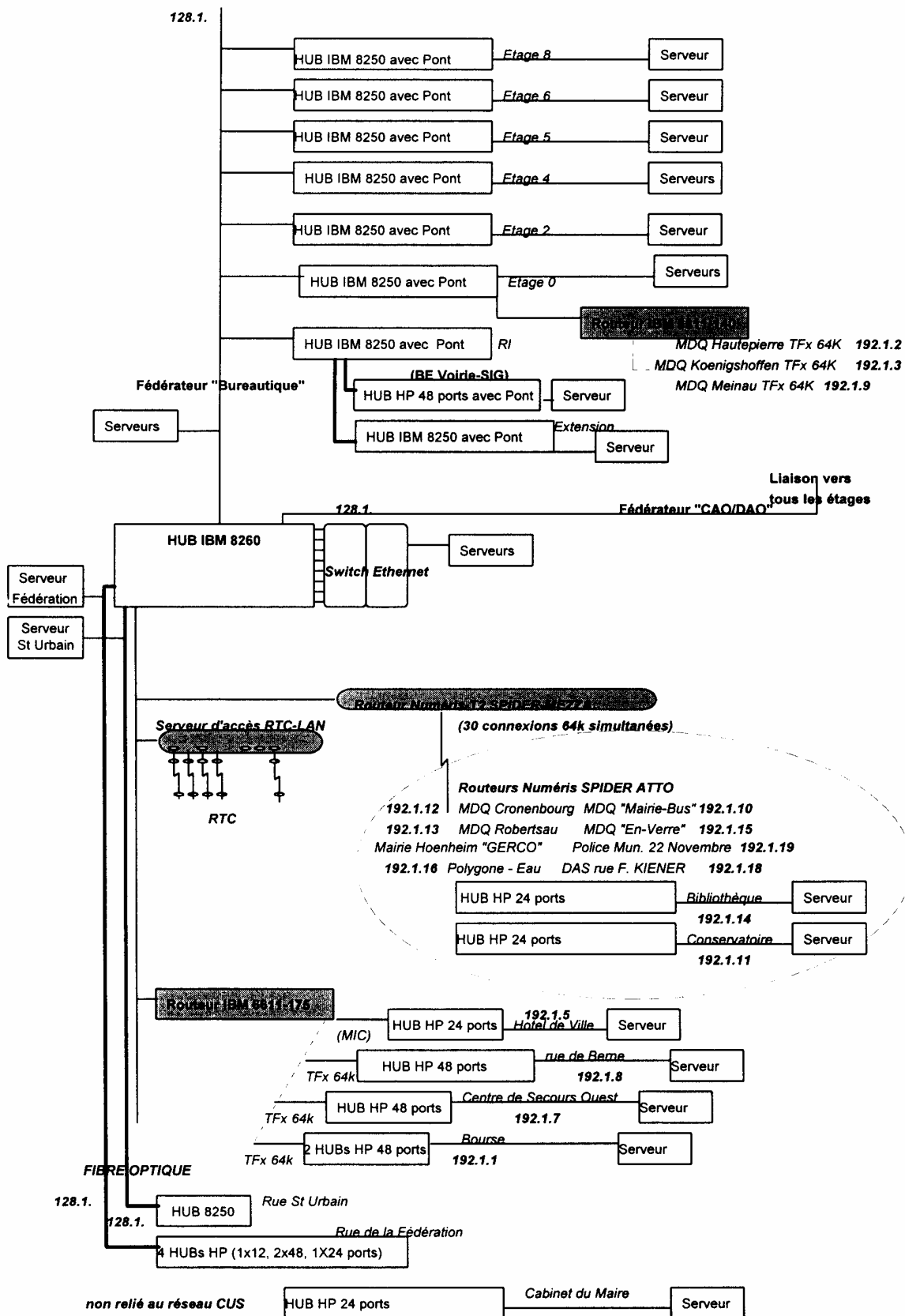
Par ailleurs, 5/6 réseaux de ce type sont mis en œuvre dans d'autres immeubles de la CUS (toujours implantés dans les différents sites géographiques de la CUS). Ces sites sont reliés au réseau fédérateur au moyen de ponts distants et de lignes spécialisées de type TRANSFIX (débits de 9600 bps à 64 kbps) et également de la fibre optique.

L'Architecture retenue pour la mise en place de la messagerie à la CUS (Mai 1995), qui tient compte des limitations actuelles des systèmes NETWARE 3.12 (nombre d'attachement d'un poste client à des serveurs NETWARE 3.12 limité à 8 au maximum, impossibilité de traiter plus de 250 connexions sur un serveur) a subi quelques adaptations par rapport à l'architecture initiale.

Schéma de principe de l'architecture du réseau pour le CENTRE ADMINISTRATIF



Situation au 1er Septembre 96



d. Organisation de la Cellule "système/réseau Mini"

La cellule "système/réseau Mini dépend du service Informatique et Réseaux de la CUS et fait partie de la Section Architecture Technique du service.

L'équipe est actuellement composée de 5 personnes

un ingénieur responsable de la Cellule "système/réseau Mini"

un technicien chargé du système AS400 ainsi que des données du gestionnaire de parc

un ingénieur réseau

deux ingénieurs chargés de l'exploitation des serveurs UNIX (Système et SGBDR)

L'organisation de l'équipe a été définie en fonction des principales missions dont elle a la charge; celles-ci concernent l'administration des logiciels systèmes sur les serveurs "minis" (matériels UNIX et AS400), du SGBDR ORACLE ainsi que du matériel Réseau faisant partie de la nouvelle Architecture du Réseau de la CUS.

Les tâches principales sont classées selon les 3 Axes mentionnés ci-dessus

1. Systèmes UNIX et AS400

- Intégration dans le réseau d'un nouveau serveur
- Définition et réactualisation du plan de sauvegarde : installation, paramétrage et mise en œuvre de ADSM (en lien avec l'ingénierie système MVS tant que le serveur ADSM est le serveur MVS)
- Gestion des sauvegardes
- Récupération des données en cas de problème
- Restauration complète d'un serveur
- Optimisation des performances
- Gestion des Incidents, assistance 3ème niveau
- Tests et mise en œuvre des produits destinés à améliorer l'exploitation des matériels et des logiciels
- Mise en œuvre et Gestion des batchs à l'aide de l'outil NETMONITOR de la Société DE31.
- Gestion de l'évolution du Système

2. SGBDR ORACLE

- Mise en exploitation d'applications
- Intégration de nouvelles Bases de données
- Administration SGBDR
- Définition du plan de sauvegarde et gestion des sauvegardes
- Gestion des Incidents, assistance 3ème niveau
- Administration et Exploitation de l'outil de supervision des SGBDR (ECOTOOLS de la société COMPUWARE)
- Tests et mise en œuvre des nouvelles versions de l'outil ECOTOOLS
Optimisation des performances
- Gestion des évolutions des produits ORACLE

3. RESEAU

- Administration de la configuration du réseau (outils NV6000 et HMP6000)
- Etude de l'expansion du réseau, c'est à dire de l'architecture et de son évolution
- Performances sur le réseau à moyen et long terme
- Etude de l'impact d'un changement sur le Réseau
- Sécurité du réseau : mise en place des outils garantissant l'intégrité, la disponibilité et la confidentialité sur le Réseau (y compris les serveurs, les SGBDR et les applications)
- Etude approfondie d'un incident Réseau
- Etude des nouvelles techniques liées au Réseau
- Mise en œuvre des nouveaux matériels sur le réseau (routeurs, hubs, etc.)