

Éléments de sécurité du système de gestion de bases de données CLIO/VAX dans un environnement réseau : conception et réalisation du processus de surveillance CIOPS

Jean-Luc Cussey

▶ **To cite this version:**

Jean-Luc Cussey. Éléments de sécurité du système de gestion de bases de données CLIO/VAX dans un environnement réseau : conception et réalisation du processus de surveillance CIOPS. Base de données [cs.DB]. 1990. dumas-00338037

HAL Id: dumas-00338037

https://dumas.ccsd.cnrs.fr/dumas-00338037

Submitted on 10 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE AGREE DE GRENOBLE (C.U.E.F.A)

U
311849

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR C.N.A.M

en

INFORMATIQUE

par

Jean-Luc CUSSEY

ELEMENTS DE SECURITE DU SYSTEME DE GESTION DE BASES DE DONNEES CLIO/VAX DANS UN ENVIRONNEMENT RESEAU CONCEPTION ET REALISATION DU PROCESSUS DE SURVEILLANCE CLIOPS

Soutenu le 20 Décembre 1990

JURY

PRESIDENT : Mr JP. MEINADIER

MEMBRES : Mr G. BEAUME
Mr J. COURTIN
Mr B. DEFUDE
Mr Y. MINAZZOLI

Je remercie,

Monsieur Jean-Pierre MEINADIER, Professeur au Conservatoire National des Arts et Métiers de PARIS, qui me fait l'honneur de présider ce jury,

Monsieur Jacques COURTIN, Professeur responsable du cycle ingénieur C.N.A.M en Informatique du centre régional de GRENOBLE, qui m'a suivi tout au long de mon cycle C,

Messieurs Georges BEAUME, Directeur de SYSECA Rhône-Alpes, et Yves MINAZZOLI, responsable du suivi du logiciel CLIO sur matériels DEC-VAX et BULL-DPS6, pour la confiance qu'ils me témoignent depuis plusieurs années, et pour l'opportunité qu'ils m'ont offerte de réaliser ce mémoire au sein de leur équipe,

Monsieur Bruno DEFUDE, Maître de conférence à l'Institut National Polytechnique de Grenoble, d'avoir bien voulu s'intéresser à mon travail et participer à ce jury,

Tout le personnel de SYSECA qui m'a aidé au cours de ce projet, parmi lequel Laurent FAIPOT, Patrice JAQUIN et Françoise PARES pour leurs conseils judicieux, et Claude CLAVEL pour son étroite collaboration durant la phase de réalisation.

Une pensée toute particulière à ma famille qui m'a soutenu pendant cette formation.

SOMMAIRE

Suite à l'accroissement considérable des configurations informatiques en réseau, est survenu un problème nouveau de sécurité d'exploitation, lié aux défauts de fonctionnement propres au réseau lui-même.

C'est pourquoi il nous est apparu indispensable, dans cet environnement qui se développe, de faire évoluer les mécanismes de sécurité du SGBD (Système de Gestion de Bases de Données) CLIO afin qu'ils puissent contrôler et gérer ce nouveau type d'anomalie.

Notre objectif a donc consisté à pourvoir le Système de Gestion de Bases de Données CLIO (variante DEC-VAX) d'une sécurité de fonctionnement optimale dans une utilisation en environnement réseau.

Cette réflexion et cette étude ont abouti à la conception et à la réalisation d'un processus de surveillance du SGBD CLIO/VAX dénommé CLIOPS.

TABLE DES MATIERES

INTRODUCTION 1

I) LE PROJET ET SON ENVIRONNEMENT 4

I.1 Syséca 5
I.2 Le marché des SGBD 5
I.3 L'équipe CLIO/VAX 7
I.4 Les clients de CLIO/VAX 7
I.5 Le projet 7
I.6 L'environnement matériel et logiciel 8

II) DEFINITION DES NOTIONS DE BD, SGBD, RESEAU, SECURITE 11

II.1 La notion de Base de Données 12
II.2 La notion de SGBD, (les fonctions qu'il assure) 12
II.3 Les réseaux informatiques (téléinformatique) 23
II.4 La sécurité des SGBD 26

II.4.1 Définition de la sécurité, les différentes sécurités 26
II.4.2 La sécurité au sens intégrité des données 28
(sûreté de fonctionnement, traitement des pannes)
II.4.2.1 La sûreté de fonctionnement
II.4.2.2 Les différents types de pannes
II.4.2.3 Mécanismes de reprises sur pannes
II.4.3 La sécurité (intégrité) dans un environnement réseau informatique 32

III) CLIOPS 34

III.1 CLIO 35

III.1.0 Historique de SOCRATE-CLIO 35
III.1.1 Présentation de CLIO 35
III.1.2 La sécurité standard de CLIO (ou CLIO et la sécurité) 37

III.2 La méthodologie utilisée pour CLIOPS 41

III.2.0 Pourquoi une méthode ? 41
III.2.1 Approche selon la méthode MERISE 41
III.2.1.0 Pourquoi MERISE ?
III.2.1.1 La méthode MERISE
III.2.1.2 Utilisation de la méthode et répartition du travail au sein de l'équipe

III.3 Conception du projet 44

III.3.0 Au départ, un problème de maintenance comme un autre 45
III.3.1 Besoins; Objectifs; Contraintes; étude théorique et de la concurrence; Orientations 47
III.3.1.1 Les besoins
III.3.1.1.1 Expression des besoins
III.3.1.1.2 Diagnostic de la situation; Evaluation des besoins
III.3.1.1.3 Formalisation du fonctionnement actuel de la sécurité CLIO et mise en évidence des limites du mécanisme actuel face aux besoins prépondérants
III.3.1.2 Objectifs
III.3.1.3 Contraintes
III.3.1.4 Etude Théorique et de la concurrence (rdb,oracle...)
III.3.1.5 Orientations choisies
III.3.2 Solutions; Choix; Conception de la solution 60
III.3.2.1 Etude Préalable
III.3.2.1.1 Présentation des solutions possibles
III.3.2.1.2 Evaluation des solutions et choix
III.3.2.1.3 La solution retenue
III.3.2.1.4 Planning
III.3.2.2 Etude Détaillée
III.3.2.2.1 Objectifs
III.3.2.2.2 Précisions et choix finaux (langage...)
III.3.2.2.3 Planning (révision)
III.3.3 Les aspects relationnels de la conception 80
III.3.4 Les difficultés rencontrées pendant la conception,les enseignements tirés 81

III.4 Réalisation de la maquette 81

III.4.1 Qu'est qu'une maquette, différence avec un prototype 81
III.4.2 Nos objectifs 82
III.4.3 Les résultats 82
III.4.4 Les difficultés rencontrées, les enseignements tirés 82

III.5 L'intégration de CLIOPS à CLIO 83

III.5.1 Objectifs 83
III.5.2 Tests, la validation du produit 84
III.5.3 Les aspects relationnels de l'intégration 84
III.5.4 Les difficultés rencontrées, les enseignements tirés 84

IV) QUEL PROCESSUS DE SURVEILLANCE POUR DEMAIN ? (QUELLE SECURITE POUR DEMAIN ?) 85

IV.1 Le CLIOPS de demain 86
IV.2 Le Processus de Surveillance de demain 88

CONCLUSION 90

ANNEXES 93

BIBLIOGRAPHIE 98

I N T R O D U C T I O N

Ce mémoire a été effectué au centre de SYSECA logiciel à MEYLAN dans l'équipe de développement et de support produit de CLIO(1) variante VAX (Digital Equipment Corporation).

Le travail réalisé consiste à pourvoir CLIO/VAX d'un mécanisme supplémentaire de sécurité afin de lui assurer un fonctionnement sans faille, dont l'élément moteur est un Processus de Surveillance.

Nous présentons tout d'abord, dans le premier chapitre, le projet et son environnement. Les aspects matériels et logiciels sont explicités de manière à introduire les concepts nécessaires à la compréhension de ce mémoire.

Le deuxième chapitre présente une étude d'ordre général sur les bases de données, les SGBD, les réseaux et leur sécurité, et définit les éléments théoriques. De plus, il présente les différents niveaux de sécurité, afin de mieux cerner le domaine d'investigation du travail réalisé, domaine de la sécurité au sens "intégrité des données" et non "confidentialité".

Le troisième chapitre aborde pleinement notre étude. Il présente de façon synthétique le SGBD CLIO et sa sécurité standard, sécurité que nous cherchons à perfectionner. Il examine ensuite notre choix méthodologique de Merise. La troisième partie de ce chapitre est consacrée à la conception de notre projet, et comporte quatre sous-parties :

* La première illustre comment l'objet de notre étude, au départ simple problème de maintenance, est devenu le projet CLIOPS.

* La deuxième examine les besoins à satisfaire, les objectifs à atteindre, et les contraintes à respecter. Elle présente la littérature existante et les travaux de la concurrence sur ce sujet. Elle propose aussi à la fin de cette étape, et compte tenu du travail effectué, une orientation possible de l'étude d'où seront issues les propositions de solutions. Il s'agit de la prise en charge et du rapport d'observation de notre méthode.

* La troisième partie, au travers de l'étude préalable, présente, puis évalue les différentes solutions possibles, et met en évidence leurs avantages respectifs. De plus, elle définit les notions employées dans la solution retenue ainsi que le planning de réalisation. Enfin, nous proposons d'exposer un extrait de l'étude détaillée afin d'illustrer nos choix finaux.

* Nous relevons enfin les aspects relationnels ainsi que les difficultés éprouvées durant cette phase de conception et les enseignements tirés.

Nous achevons ce troisième chapitre en livrant l'expérience enrichissante que constituent le maquettage et la réalisation technique de notre projet, tout en relevant les difficultés rencontrées.

Enfin, le dernier chapitre essaie d'entrevoir l'avenir. Nous cherchons à présenter la sécurité du futur ainsi que notre CLIOPS de demain.

Un retour sur le chemin parcouru et le travail réalisé nous permet de faire le bilan critique de notre projet, et de rappeler, en conclusion, tout l'enrichissement qu'il nous a apporté.

(1) CLIO est un produit de SYSECA logiciel

CHAPITRE I

LE PROJET ET SON ENVIRONNEMENT

I.1 SYSECA

SYSECA est une Société de Service et d'Ingénierie en Informatique, créée en 1966. Il s'agit d'une filiale de THOMSON CSF et fait partie du pôle informatique THOMSON. Son fondateur fut Monsieur Pierre THELLIER. Depuis la fusion récente de SYSECA avec SODETEG TAI, Monsieur Chevrel est le nouveau Président Directeur Général et Monsieur Bertin le Directeur Général.

L'effectif de Syséca dépasse actuellement les 1900 personnes pour un chiffre d'affaires de plus de 9OO MF, ce qui place cette entreprise en 10ème position dans le classement des SSII [01INFO89a].

Son siège se situe à SAINT-CLOUD, mais SYSECA compte de nombreuses implantations dans toute la France (Brest, Grenoble, Lille, Lyon, Nantes, Rennes, Toulon, Toulouse), et également des filiales à l'étranger.

Les domaines d'activités de SYSECA sont les suivants :

* génie informatique
* gestion
* messageries et télécommunications
* systèmes industriels
* systèmes temps réel

SYSECA SA est composée de trois filiales, qui sont SYSECA TEMPS REEL, SYSECA LOGICIEL et SYSECA TAI.

L'agence de SYSECA implantée sur la ZIRST de Meylan, fait partie de SYSECA LOGICIEL. Elle emploie actuellement environ 80 personnes. L'activité principale de SYSECA Meylan est d'assurer le développement de trois progiciels (de la division génie informatique) et le suivi de la clientèle (installation, formation, assistance).

Les trois progiciels développés sont à ce jour :

* CLIO : S.G.B.D (installé sur plus de 900 sites).
* SACADO : gestion et archivage documentaires (installé sur plus de 150 sites).
* MUST : système de gestion d'informations des années 90, en cours de réalisation.

I.2 LE MARCHE DES SGBD

On peut en quelques mots décrire le marché du SGBD. [01INFO89b] nous indique que les SGBD traditionnels représentent encore l'essentiel d'un marché mondial évalué selon Dataquest, à près de 2,5 milliards de dollars de chiffre d'affaires annuel . De plus, le marché des SGBD sur gros systèmes représente, selon Input, un volume d'affaires annuel de 1,2 milliards de dollars, et devrait croître plus lentement que le marché dans son ensemble.

En effet, une croissance de 12% est prévue d'ici 1992. Les chiffres de [01INFO89d] précisent ces informations : selon Input, le marché mondial des SGBD représentait plus de 2,2 milliards de dollars en 1987, qui se divisaient respectivement en 580, 510 et 1150 millions de dollars pour les applications sur PC, mini-informatique et grands systèmes. En 1992, ces chiffres auront plus que doublé puisque les analystes d'Input évaluent le marché des SGBD à plus de 5,8 milliards de dollars. Mais la configuration du marché se sera alors profondément modifiée : la croissance rapide des SGBD pour des applications en micro et mini-informatique, dont la part passe de 26% à 31% pour les premières et de 23% à 33% pour les secondes, se fera au détriment des applications grands systèmes, qui chutent de 51% à 36% sur cette période. Le schéma suivant, issu de [01INFO89c], illustre cette tendance et donne les chiffres correspondant à l'Europe:

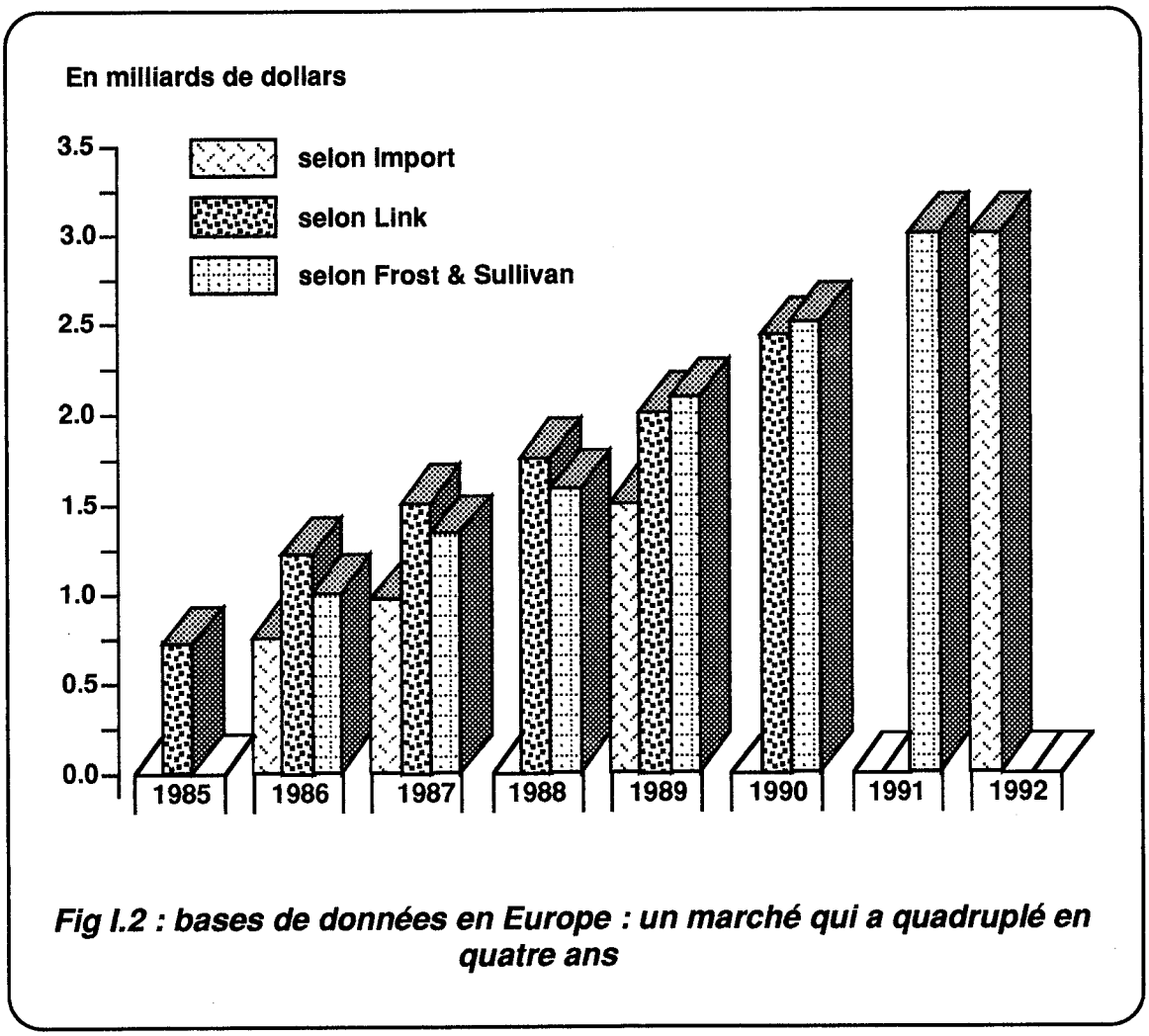


Fig I.2 : bases de données en Europe : un marché qui a quadruplé en quatre ans

L'avenir est donc propice, comme l'illustre le titre de l'article de [01INFO89b] : "Les SGBD décollent, une croissance annuelle d'environ 25% d'ici 1991 devrait donner des ailes aux développeurs de SGBD". Mais les auteurs ajoutent dans leur article: "il faut dire que le ticket d'entrée sur le marché des SGBD est particulièrement onéreux : plus de 20% du chiffre d'affaires doit être consacré au financement de la recherche et du développement..." (ce qui explique peut-être notre projet).

I.3 L'EQUIPE CLIO/VAX (le cadre du mémoire)

Je suis intégré depuis avril 1988 dans l'équipe CLIO VAX, qui assure le développement, le support produit du SGBD CLIO variante Vax (Digital), l'installation des nouveaux sites et l'assistance technique. J'anime également le Cercle des Utilisateurs CLIO VAX, qui se réunit quatre fois par an. De plus, j'ai été chargé de mener à bien un projet d'évolution du mécanisme de "journalisation après" de CLIO. Il permet à CLIO/VAX un fonctionnement ininterrompu et offre un système de reprise automatique sur incident mécanique (perte d'unité centrale ou de disque support de CLIO) pour le contrôle d'une chaîne de retraitement de déchets nucléaires.

I.4 LES CLIENTS DE CLIO/VAX

Il me semble important de présenter en quelques lignes les clients du SGBD CLIO/VAX, car cela peut aider à bien comprendre l'attente et l'exigence de ces entreprises. Actuellement, on se rapproche de la centaine de CLIO/VAX installés. Les clients appartiennent à des domaines d'activités variés: Aérospatiale, Télécommunications, Chambres de commerces, Nucléaire, Centrales d'achats, Banques, Industries pharmaceutiques, Industries lourdes ...

Les clients relèvent, pour la majorité, d'entreprises privées ou de sociétés nationales de haute technologie. Aussi, ne peuvent-ils pas se permettre de ressaisir des informations perdues, car cela correspond à une lourde perte financière, et à une perturbation inadmissible de l'activité de l'entreprise. La notion de sécurité et de "fonctionnement ininterrompu" prend ici tout son sens...

I.5 LE PROJET

C'est dans ce cadre (maintenance CLIO/VAX, contacts directs avec les clients par le biais des interventions, cercle des utilisateurs et projet sur la sécurité par JNLA) que j'ai été sensibilisé aux problèmes de sécurité des SGBD, et plus particulièrement à celle de CLIO.

Compte tenu des demandes des clients et de l'évolution du marché en matière d'ordinateur en environnement réseau (cf III.1), j'ai été amené à proposer une évolution du produit CLIO.

Le projet réalisé consiste à pourvoir CLIO/VAX d'un mécanisme supplémentaire de sécurité afin de lui assurer un fonctionnement sans faille dans un environnement réseau, et de permettre à CLIO de délivrer des informations sur le contexte de travail des utilisateurs. L'élément moteur est un Processus de Surveillance dénommé CLIOPS.

Cette évolution a été effectuée dans le cadre d'une étude interne, et a été intégrée à une nouvelle version CLIO V4+ actuellement en field-test, et prochainement livrée en clientèle pour une diffusion générale prévue au début de l'année 1991.

Durant ce mémoire, je suis tout naturellement resté dans l'équipe CLIO/VAX (cf III.2.1.2).

I.6 L'ENVIRONNEMENT MATERIEL ET LOGICIEL

Nous travaillons sur matériel Digital Equipment Corporation (DEC), muni du système d'exploitation multi-utilisateurs VMS :

* MICROVAX 3600 : 3 disques (1,2 Giga Octets), mémoire 32 Méga Octets.
* VAX 8250 : 3 disques (1,8 Giga Octets), mémoire 8 Méga Octets.

Les langages utilisés pour la programmation de CLIO sont Pascal, C et le macro langage MACR0-32 de VMS pour la programmation en langage assembleur.

I.6.1 DIGITAL

DIGITAL (Digital Equipment Corporation) est le deuxième constructeur de matériel informatique dans le monde [O1INFO89c] et le "leader" mondial en matière de réseau informatique. Cette société a été créée en 1957, et est apparue en France en 1965. En France elle se place actuellement au 3ème rang des constructeurs informatiques derrière IBM et BULL [01INFO89a].

Le Président Général de Digital Corporation est Kenneth H. Olsen. Monsieur Ferreboeuf est Président du comité de direction de Digital France et l'entreprise emploie actuellement 4300 personnes [DICSI90].

I.6.2 VMS

VMS (Virtual Memory System) est le système d'exploitation de Digital pour les matériels de la gamme VAX. C'est un système d'exploitation multi-utilisateurs pouvant traiter simultanément plusieurs applications.

Nous avons, pour notre part, effectué nos développements en utilisant les versions VMS 4.7 et 5.2.

De plus, VMS offre un mode de fonctionnement identique sur toute la gamme Digital VAX, tant sur le plan logiciel que sur le plan matériel. Ceci n'empêche pas un large éventail de prix et de puissances: de 0,8 à plus de 100 vups (1 Vax Unit Performance = 1,2 mips).

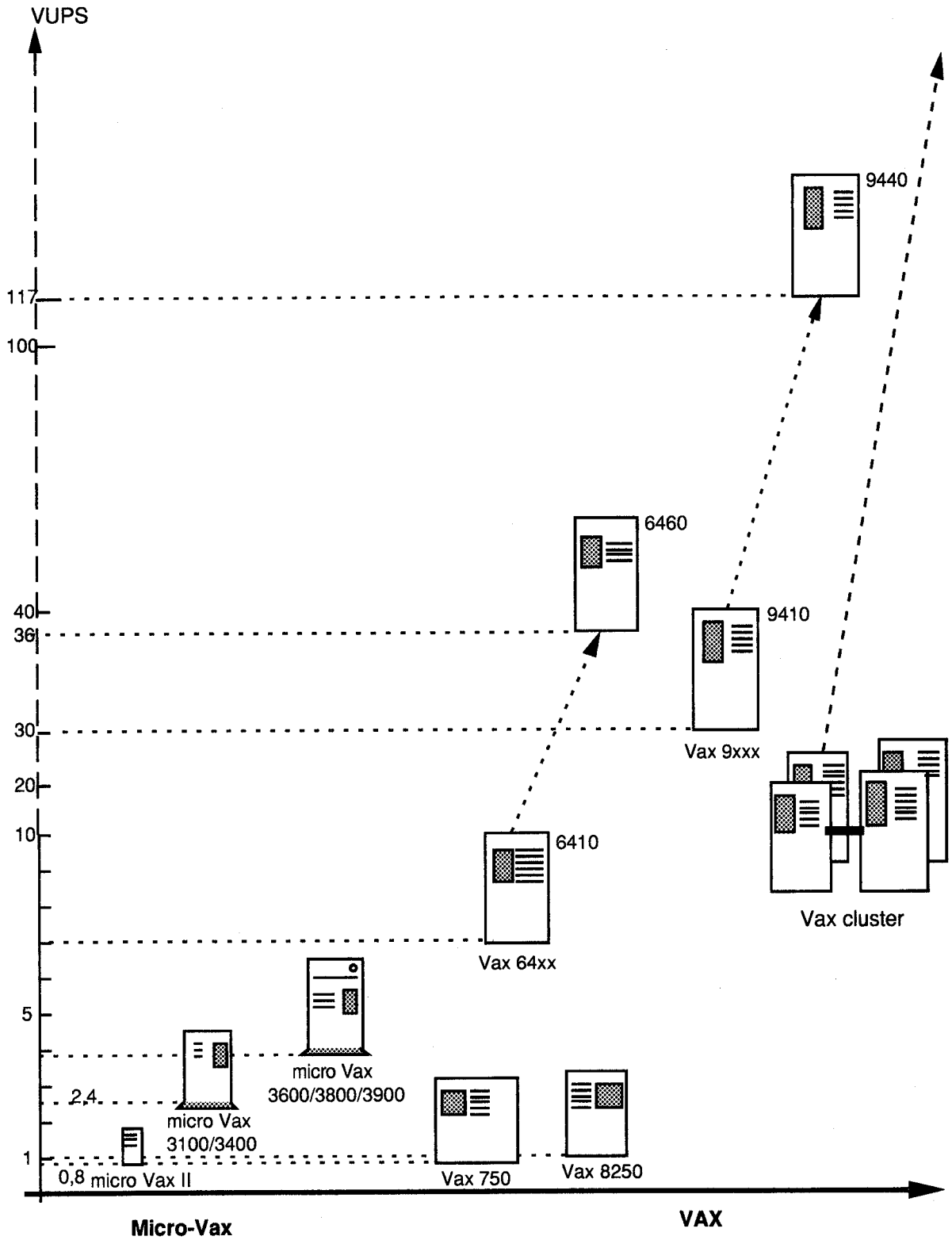


Fig I.6.2 : Performance des familles MICRO-VAX et VAX de DIGITAL

VMS se caractérise essentiellement par:

- son système de gestion de la mémoire virtuelle qui permet en théorie une non limitation du nombre de processus concurrents
- ses communications inter-tâches qui peuvent s'effectuer à travers des fichiers communs, des zones de mémoire communes, des boîtes aux lettres (mailboxes) ou des sémaphores.
- sa gestion des systèmes multiprocesseurs et une sécurité de très haut niveau (VMS a reçu, dans sa version 4.3, l'agrément du ministère de la Défense des Etats-Unis (classe C2)).

VMS intègre toute une panoplie d'outils fort appréciés des développeurs. Ceux-ci procurent un environnement complet de mise au point de logiciels, du type utilitaire, dans un environnement intégré. Mais il existe aussi des utilitaires de type "boîte à outils", que l'on peut acquérir en supplément d'un environnement déjà très riche.

Les principaux outils que nous avons été amené à utiliser pendant le projet sont :

- help : aide possible à tous niveaux. Il peut être invoqué à partir d'un utilitaire ou directement sous VMS
- pas : compilateur pascal
- EDT, TPU : éditeur pleine page bi-fenêtres
- debug : le dévernisseur (debugger) de VMS: utilitaire de mise au point symbolique multifenêtres, indépendant du langage de programmation utilisé
- link : éditeur de liens
- monitor : utilitaire de surveillance du système ou d'un processus particulier.

CHAPITRE II

DEFINITION DES NOTIONS DE BD, SGBD, RESEAU ET SECURITE

Avant de parler du SGBD CLIO et du processus de surveillance CLIOPS, il convient de définir les termes de "Bases de Données", de "Système de Gestion de Base de Données", de "Sécurité" et de "Réseau".

II.1 LA NOTION DE BASE DE DONNEES

Plusieurs définitions s'offrent à nous:

Pour le dictionnaire "petit Robert" :
<<Base de données (de l'anglais DATA BASE): ensemble de données logiquement reliées entre elles et accessibles au moyen d'un logiciel spécialisé>>.

[MINI89] nous livre celle du journal officiel :
<<Base de données: ensemble de données organisées en vue de son utilisation par des programmes correspondant à des applications distinctes et de manière à faciliter l'évolution indépendante des données et des programmes (journal officiel du 17 janvier 1982)>>.

[ADIB83] nous indique qu'<<une base de données est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective>>.

Nous avons aussi relevé celle de [FLOBOU86]:
<<Une base de données représente un ensemble de données de l'entreprise mémorisé par un ordinateur et dont l'organisation est régie par un modèle de données>>.

II.2 LA NOTION DE SGBD

Envisageons maintenant la définition du concept contenu dans le titre de notre étude: "Système de Gestion de Base de Données".

II.2.1 DEFINITION DU SGBD ET DE SES FONCTIONS

Une première approche nous est livrée par la définition de [FLOBOU86] : <<Un SGBD représente un ensemble coordonné de logiciels qui permet de décrire, mémoriser, manipuler, interroger, traiter les ensembles de données constituant la base>>.

[ADIB83] le décrit ainsi: <<le logiciel qui permet à un utilisateur d'inter-agir avec une base de données est un système de gestion de base de données (SGBD). Il permet principalement d'organiser les données sur le support périphérique et fournit les procédures de recherche et de sélection de ces mêmes données. Pour aboutir à ce résultat, l'utilisateur décrit, en termes abstraits, ce qu'il souhaite faire sur les données, laissant le soin au système d'effectuer les tâches de recherche, en fonction de la représentation et de l'organisation des données sur les supports physiques.>>

Un SGBD doit en outre assurer les fonctions suivantes:

* description logique (perception de la base par l'utilisateur) et physique (organisation sur le support physique interne) des données
* utilisation des données (interaction de l'utilisateur avec la base de données sous forme de dialogue)
* intégrité des données et des règles de contrôle
* confidentialité des informations (droits d'accès)
* gestion des concurrences d'accès
* sécurité complète de fonctionnement pour tous les types d'interruption.

[AKOKA84] nous explique qu'il vaut mieux définir le SGBD en précisant certaines fonctions qu'il doit remplir, afin de déterminer si un système est réellement un SGBD:

* intégration des données
* séparation entre moyens de stockage physique des données et la logique des applications
* contrôle unique de toutes les données afin de permettre à plusieurs utilisateurs un emploi simultané
* possibilité d'utiliser des structures de fichier et des méthodes d'accès complexes
* facilités pour le stockage, la modification, la réorganisation, et la consultation des données
* contrôles de sécurité afin d'empêcher l'accès illégal à certaines données
* contrôles d'intégrité pour prévenir une modification indue des données
* compatibilité avec les principaux langages de programmation, les programmes sources existants, et les données extérieures à la base.

II.2.2 LES MODELES DE BASE DE DONNEES, LES DIFFERENTS SGBD

II.2.2.1 La notion de "modèle de donnée"

Après avoir expliqué les notions de "Base de données", de "SGBD" et ses fonctions, nous pouvons compléter ces définitions, par un schéma qui permet de visualiser les différents niveaux d'une base de données selon que l'on observe celle-ci avec une approche utilisateur ou sous un aspect plus technique.

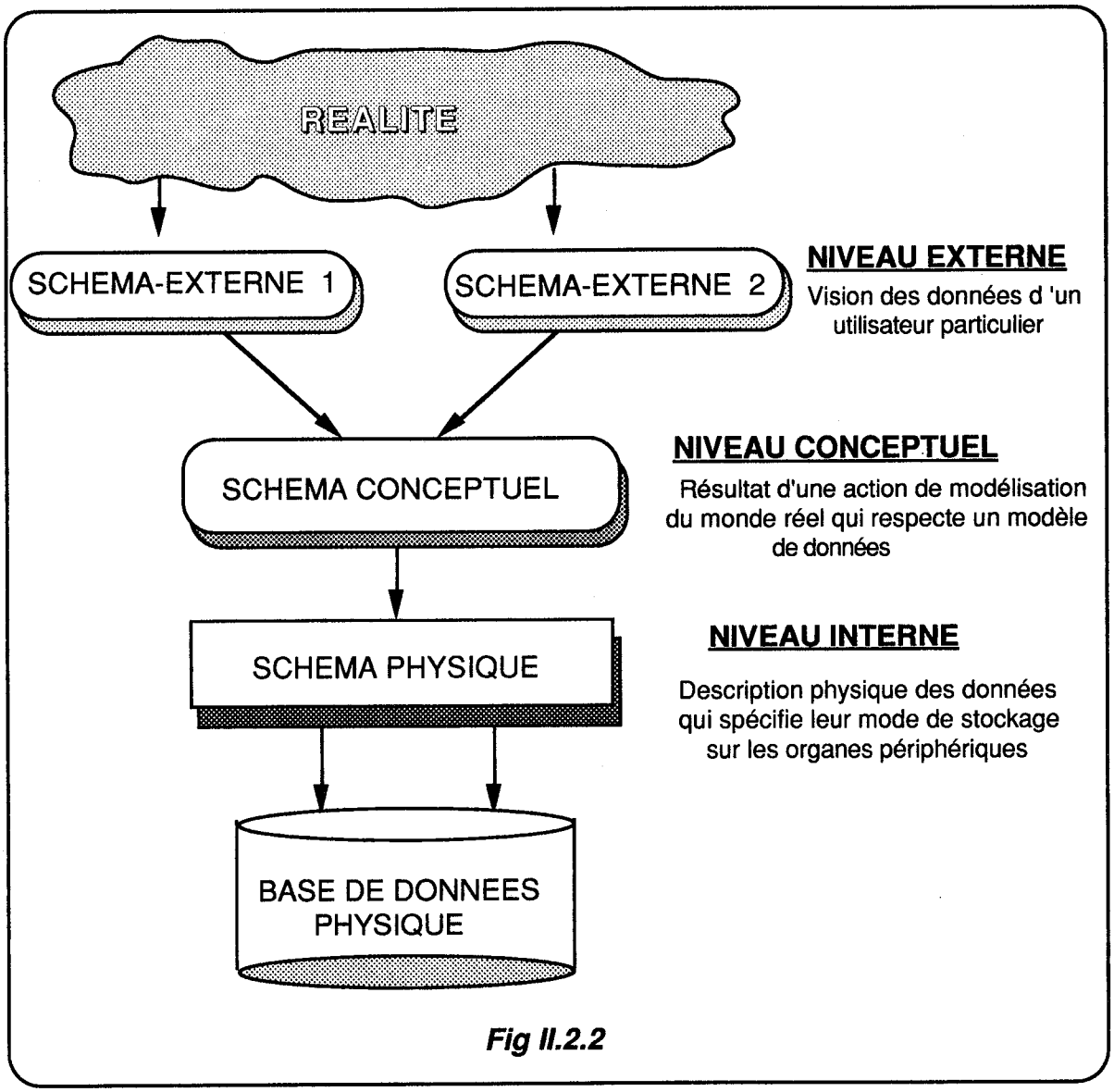


Fig II.2.2

Ce schéma fait apparaître les notions de "modèle" et de "modélisation", déjà abordées dans la définition des bases de données de [FLOBOU86]. On peut développer ce concept fondamental qui permet de différencier les divers types de SGBD :

Pour [AKOKA84] <<un modèle de données est un outil pour représenter l'organisation logique des données. Il est utilisé pour décrire et enregistrer l'interprétation du réel sur un support physique>>.

[MIRA86] nous précise que le modèle conceptuel utilise deux notions de base: <<l'objet et le lien>>.

[AKOKA84] nous différencie les modèles de données par la façon dont ils représentent les relations entre les données : <<il en existe actuellement trois de répandus:

* le modèle hiérarchique
* le modèle réseau
* le modèle relationnel.>>

II.2.2.2 Les différents SGBD (et leur modèle)

Le concept de base de données est né dans les années 1960, avec la prise de conscience par les développeurs d'applications des contraintes qu'imposait l'utilisation des méthodes de développement traditionnelles (approche fichiers de données) :

- redondance de données
- lourdeur de programmation
- difficulté à maintenir la cohérence du système d'informations
- difficulté d'évolution.

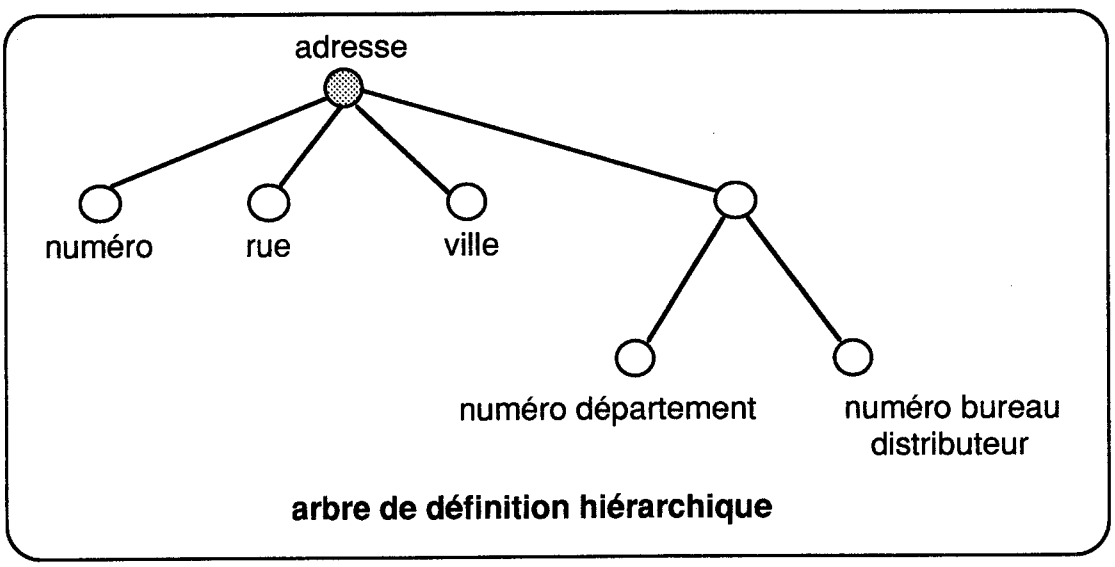
Les premiers SGBD apparus sur le marché ont tenté de répondre à ces problèmes. Il s'agit des SGBD de type hiérarchique utilisant un modèle de même type.

a) SGBD Modèle hiérarchique : "un arbre ordonné"

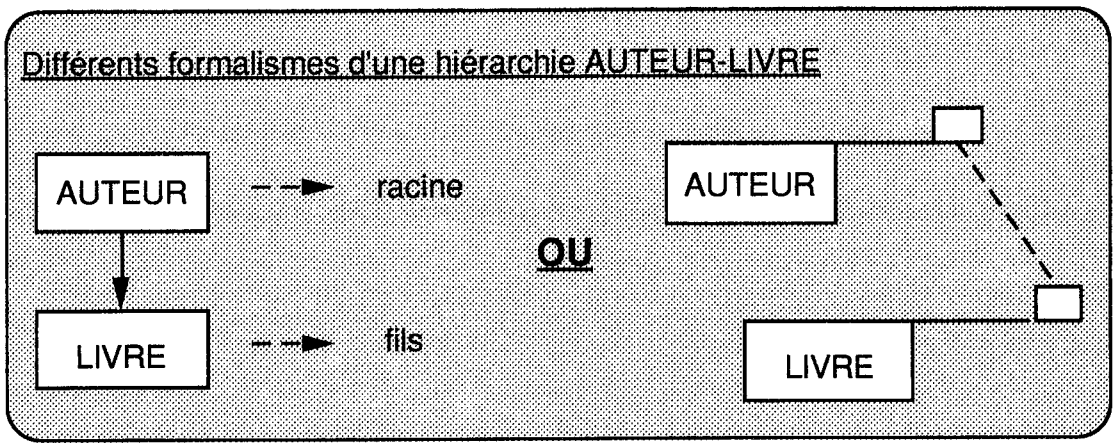
Il a été créé dans les années 1960.

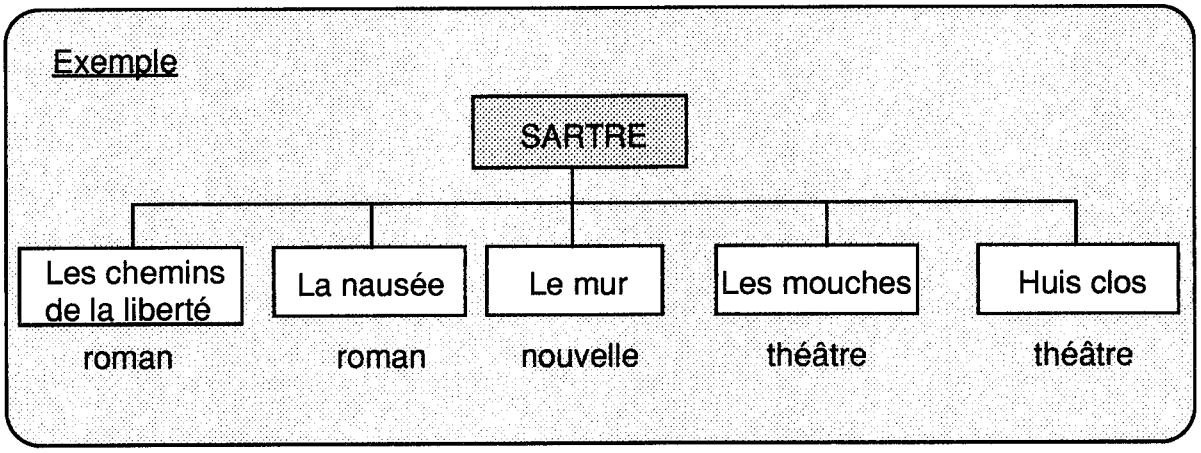
Définition :

Elle nous est donnée par [ADIB83] : <<A l'aide du modèle hiérarchique, le schéma conceptuel peut être visualisé sous forme d'un graphe arborescent dont les noeuds correspondent aux classes objets et les arcs entre les noeuds aux associations. Un tel graphe possède un noeud racine et les autres noeuds sont appelés fils, petit-fils.>>



arbre de définition hiérarchique





Exemples de SGBD hiérarchiques :

IMS, SYSTEM 2000, TDMS, MARK IV, RFMS

Conclusion: avantages et inconvénients:

L'organisation hiérarchique reflète souvent le besoin de l'utilisateur, et la simplicité du modèle en fait l'un des plus utilisés parmi les SGBD commercialisés.

Mais cependant, beaucoup d'inconvénients sont relevés. Parmi eux :

- les relations du type "plusieurs à plusieurs" (N:M) ne peuvent être facilement représentées
- il nécessite un espace de stockage important
- il n'existe qu'un seul chemin d'accès possible aux données
- la modification du schéma entraîne la restructuration de la base, et par là un coût prohibitif des traitements ad hoc (non prévus)
- l'indépendance logique est très réduite; la structure du schéma doit refléter tous les besoins des applications
- l'interface utilisateur non procédurale est inexistante.

b) SGBD Modèle réseau : "un réseau d'entités et de pointeurs logiques"

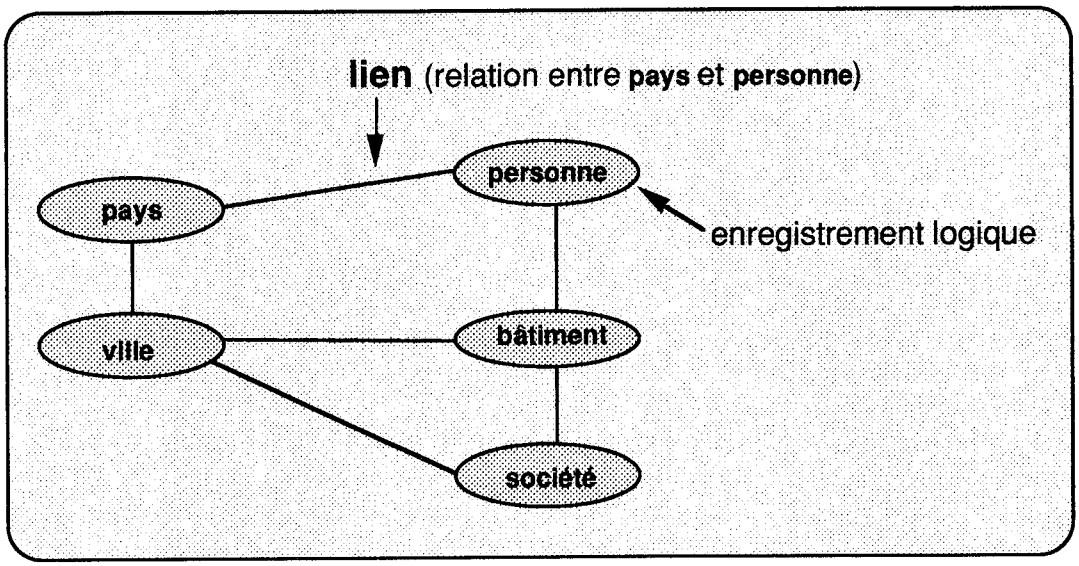
Le modèle en réseau a ses origines (1963) dans le développement d'IDS et d'APL IDS ("Integrated Data Store") développé par Bachmann et ses collègues de la "Général Electric Company". Sur cette base, le groupe CODASYL (Conférence On DAta SYstem Language) démarre en 1968 l'étude d'une extension de Cobol pour manipuler les bases de données, et fait en 1969 ses premières recommandations concernant la syntaxe et la sémantique :

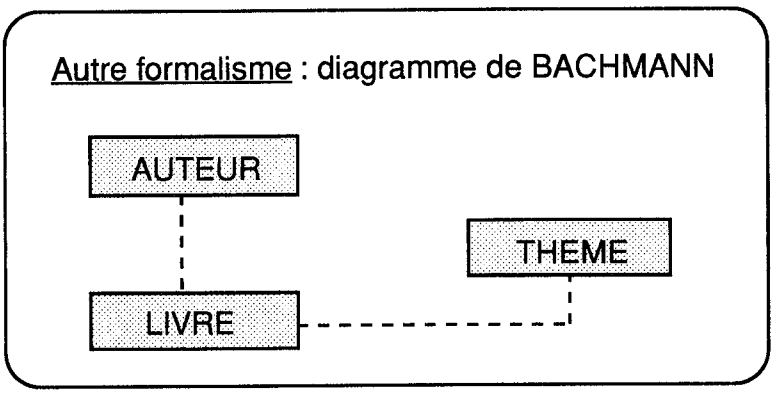
- du langage de définition de données pour décrire un schéma en réseau
- du langage de manipulation de données (qui est un enrichissement de Cobol)

Avec la création de lien logique entre des objets différents et la possibilité de plusieurs chemins d'accès à une donnée, les SGBD réseaux lèvent les principaux inconvénients du modèle hiérarchique.

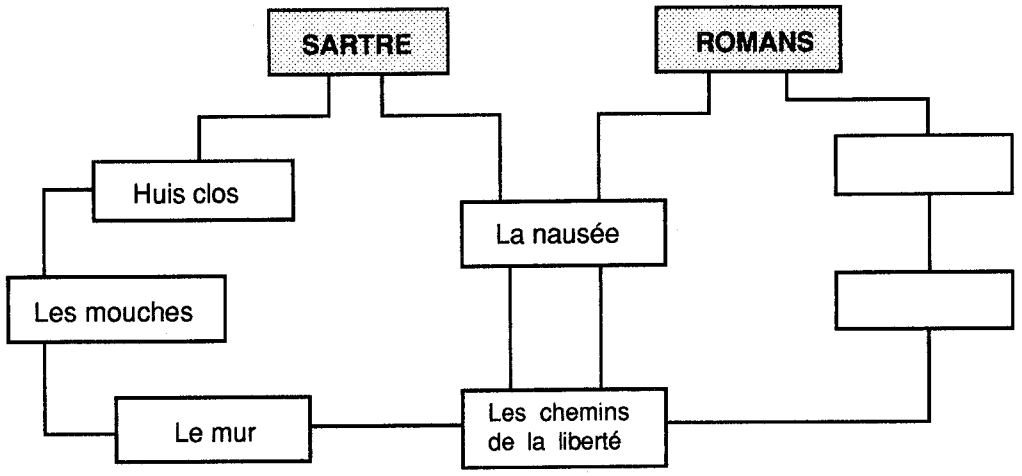
définition:

Le schéma est présenté sous forme d'un graphe, d'un réseau (d'où le nom du modèle) connectant les entités entre elles à l'aide de pointeurs logiques.





Exemple



On peut donner quelques dates:

1959 : COBOL
1963 : IDS (réseau)
1966 : recommandations du groupe DBTG (Data Base Task Group)
1969 : premier rapport CODASYL
1970 : industrialisation de Socrate sur IRIS 50
1971 : deuxième comité DBTG (LDD,LMD)
1978 : IDS II
1984 : CLIO

Exemples de SGBD réseau :

DBMS, IDMS, TOTAL, EDMS, IDS II, SOCRATE-CLIO

Conclusion: avantages et inconvénients:

Le modèle réseau est le fruit logique de l'évolution du modèle hiérarchique; il remédie à certains inconvénients car, contrairement au précedent, la représentation graphique ne comporte aucune limitation. Il définit plus particulièrement des notions d'enregistrement logique et de lien, permettant l'élimination des données redondantes et la création de chemins multiples vers une donnée à travers des liens logiques. Il permet donc la représentation naturelle des liens maillés. De plus, il améliore la rapidité et la sécurité, et augmente la productivité des informaticiens.

Mais l'on peut noter néanmoins quelques inconvénients:

- l'absence d'indépendance vis-à-vis des stratégies d'accès; les structures sont pensées en terme d'accès aux données (confusion du modèle conceptuel et du modèle des chemins d'accès)
- la procéduralité importante des langages de manipulation de données car l'utilisateur doit "naviguer" dans le réseau logique constitué par les enregistrements et les chaînes de pointeurs
- la nécessité d'un schéma de données stabilisé.

c) SGBD Modèle relationnel : "une vue tabulaire des données"

En 1969, E.F CODD, mathématicien du centre de recherche d'IBM à San José (Californie), publie un article jetant les bases du modèle relationnel [MIRA86] :

<<Il constitue une approche de la description et de la manipulation logique des données trés différentes de celles des modèles hiérarchiques et réseaux.>>

définition :

[AKOKA84] : <<Le modèle relationnel envisage la base de données comme un ensemble de tableaux à deux dimensions appellés relations.>>

[ADIB83] nous indique que <<ce modèle est basé sur la notion mathématique de relation suivant laquelle une relation est un ensemble de n-uplets>>.

Par exemple : l'association INSCRIPTION pourra être représentée comme une relation ayant même nom et construite sur trois ensembles ETUDIANT - ENSEIGNEMENT - ANNEE. On écrira formellement :

INSCRIPTION = [(x,y,z)/l'étudiant x est inscrit à l'enseignement y de l'année z]

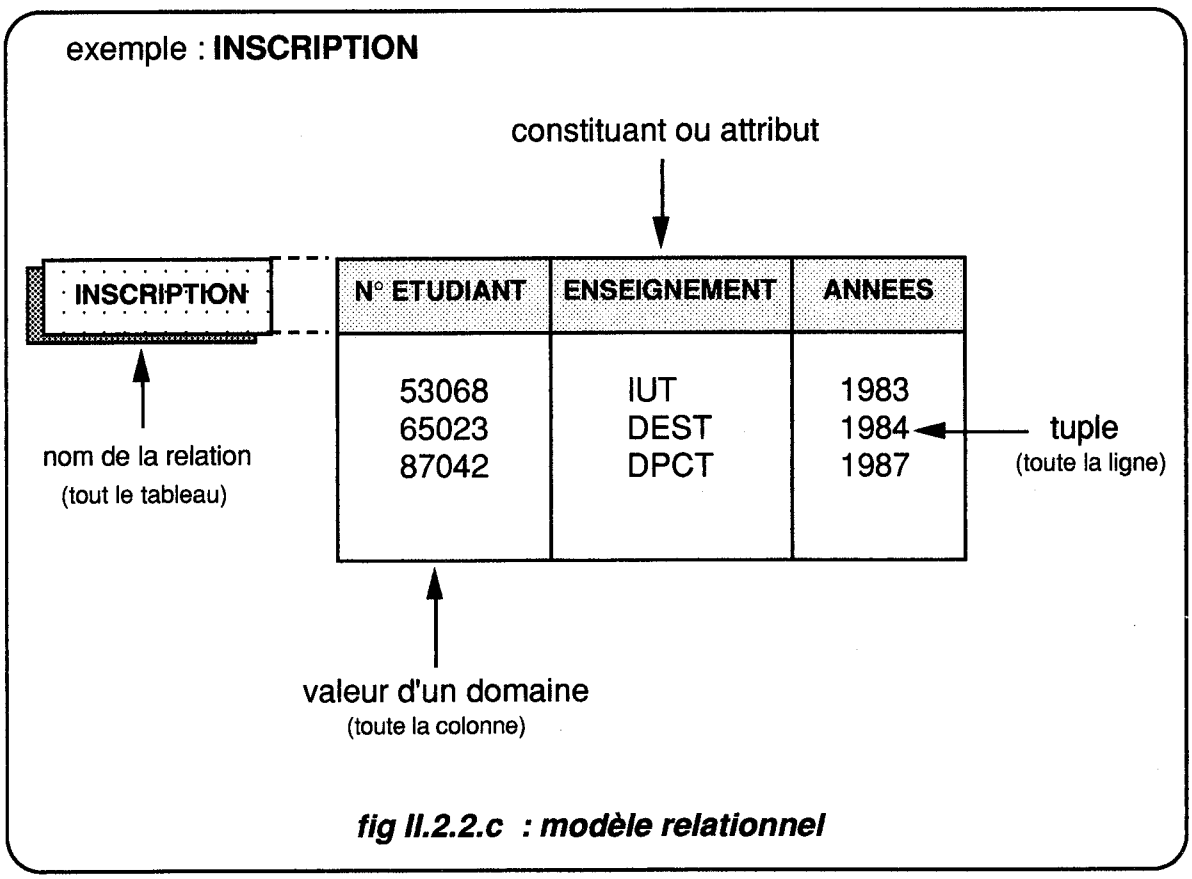


fig II.2.2.c : modèle relationnel

On peut citer quelques dates importantes pour les SGBD relationnels :

1969-70 : E.F CODD conçoit le modèle relationnel
1972 : forme stabilisée du relationnel
1975 : naissance des SGBD: SYSTEM R, MRDR
1976 : définition du langage SQL (Structural Query Langage)
1979 : première installation d'Oracle
1986 : SQL devient un standard

Exemples de SGBD relationnels :

DB2, INGRES, SABRE, DATACOM/DB, FOCUS, NOMAD2, ORACLE, RAPPORT, UNIFM, CLIO-V6, DMS2, IDMS/R, ULTRA ...

Conclusion : avantages et inconvénients:

Malgré sa complexité théorique, le modèle relationnel a pour but de simplifier les structures de données utilisées dans les modèles hiérarchiques et réseaux. Il semble connaître actuellement un succès commercial. On peut citer parmi les avantages reconnus du modèle relationnel:

- l'indépendance de l'utilisateur vis-à-vis des structures logique et physique des données et de leurs stratégies d'accès
- la puissance et l'uniformité de représentation
- la facilité et la clarté de description et d'utilisation de la base de données (l'utilisateur n'a plus à s'occuper des stratégies d'accès comme dans le modèle réseau).

Les inconvénients de ce modèle résident dans:

- le problème de performance (temps d'accès aux informations plus importants que ceux des bases de données hiérarchiques et réseaux)
- la perte d'un certain degré d'indépendance logique lors de la normalisation
- le manque de symétrie d'une relation dû à une distinction entre entités et objets. Par exemple, pour le commun des mortels, une adresse est un objet de l'entité personne; c'est l'inverse pour le postier, pour qui une personne est un objet de l'entité adresse
- le fait que toutes les relations ne jouent pas le même rôle (certaines décrivent des entités statiques, d'autres des liens)
- la perception des relations entre les données sous forme de tableau, interdit toute représentation naturelle du monde réel.

Ce modèle semble particulièrement approprié à un univers dynamique, dont l'utilisation est mal connue a priori et ne nécessite pas de grandes performances.

d) Les SGBD du futur, évolutions actuelles:

Comme nous l'avons montré dans nos paragraphes précédents, les SGBD évoluent par le biais de leur modèle mais, dans leur globalité, ils semblent répondre aux besoins actuels des applications de gestion. Néanmoins, des demandes croissantes apparaissent, notamment en CAO ou CFAO, ainsi que des concepts nouveaux véhiculés par l'intelligence artificielle qui mettent en évidence les lacunes des SGBD existants.

Cette évolution n'étant pas l'objet principal de notre étude, il me semble néanmoins opportun de la présenter car nous y reviendrons dans le dernier chapitre de notre mémoire ("quelle sécurité pour demain ?"), les bases en seront alors jetées.

On peut noter six lacunes des SGBD actuels:

[GARVAL85] en soulève trois:

<< - les nouveaux types de données qualifiées de "multi-médias" (textes, images, sons, résultats de mesure, documents ...) ne peuvent être facilement traités
- les SGBD actuels ne sont guère capables de raisonner afin de déduire de nouvelles données à partir des données existantes et des lois générales connues
- les SGBD sont de plus en plus conviviaux mais il faut encore améliorer les interfaces, tant pour les utilisateurs que pour les développeurs. >>

[GARD83] nous en livre une quatrième:

<<La décentralisation des systèmes informatiques se heurte au problème de la répartition des données>>.

[FLOBOU86] nous indique une cinquième carence:

<<Le modèle relationnel est très satisfaisant pour les données élémentaires mais peu pour les relations entre entités et l'on peut dire le contraire du modèle réseau>>.

[FLOBOU86], [ADIB83] et [MIRA86] posent enfin le problème de la performance des SGBD relationnels:

<<De plus en plus de SGBD sont relationnels mais subsiste le problème de leurs performances>>.

<<Dans le cadre des SGBD relationnels l'aspect performance est un facteur déterminant>>.

<<Le principal problème à résoudre dans la mise en oeuvre des SGBD relationnels est le problème de performance>>.

Dans le but de satisfaire ces besoins, de nouveaux SGBD se dessinent :

- SGBDOO (SGBD Orienté Objet) pour traiter des objets complexes (ex: O2, IRIS, GBASE, VBASE, GEMSTONE)

- SGBD Déductif , donnant aux SGBD des possibilités de déduction en intégrant au sein du SGBD des mécanismes intelligents d'inférence, par interfaçage avec un langage d'intelligence artificielle (PROLOG, LISP). Par exemple, le couplage de SUPPRA avec RDLM ou de CLIO avec ART, l'intégration à l'intérieur du SGBD d'un support complet de règles permettant de définir des prédicats récursifs (ex : POSTGRES).

- intégration ou interfaçage d'outils de type tableur, grapheur, générateur d'applications...

- SGBD réparti assurant globalement la cohérence, la sécurité, l'intégrité et la fiabilité des données géographiquement disséminées sur plusieurs systèmes (ex: CLIO, EMPRESS, INGRES*STAR, ORACLE*STAR, ADABAS*NET, SUPRA*NET, DB2, SYBASE).

- apparition ou réactualisation de nouveaux modèles
  [FLOBOU86] : <<c'est pour essayer de faire une synthèse de deux approches que CHEN propose en 1976 le modèle entité relation (qui revient à la mode) qui se rapproche des besoins réels de l'entreprise>>
  [MBD89] : <<le modèle de données objet tente de dépasser les contraintes du modèle relationnel et l'ambigüité du modèle entité-relation en s'appuyant sur le paradigme d'objet ...>> Ce modèle a été conçu pour être utilisé dans le cadre d'une méthode d'analyse (MERISE,AXIAL), et se situe dans la continuité de CODASYL, du relationnel et du modèle entité relation.

- machines base de données intégrant le maximum de fonctions au niveau "hard", déchargeant ainsi l'ordinateur de ses tâches de gestion de données (ex: IDM).

Le schéma suivant synthétise le monde du SGBD:

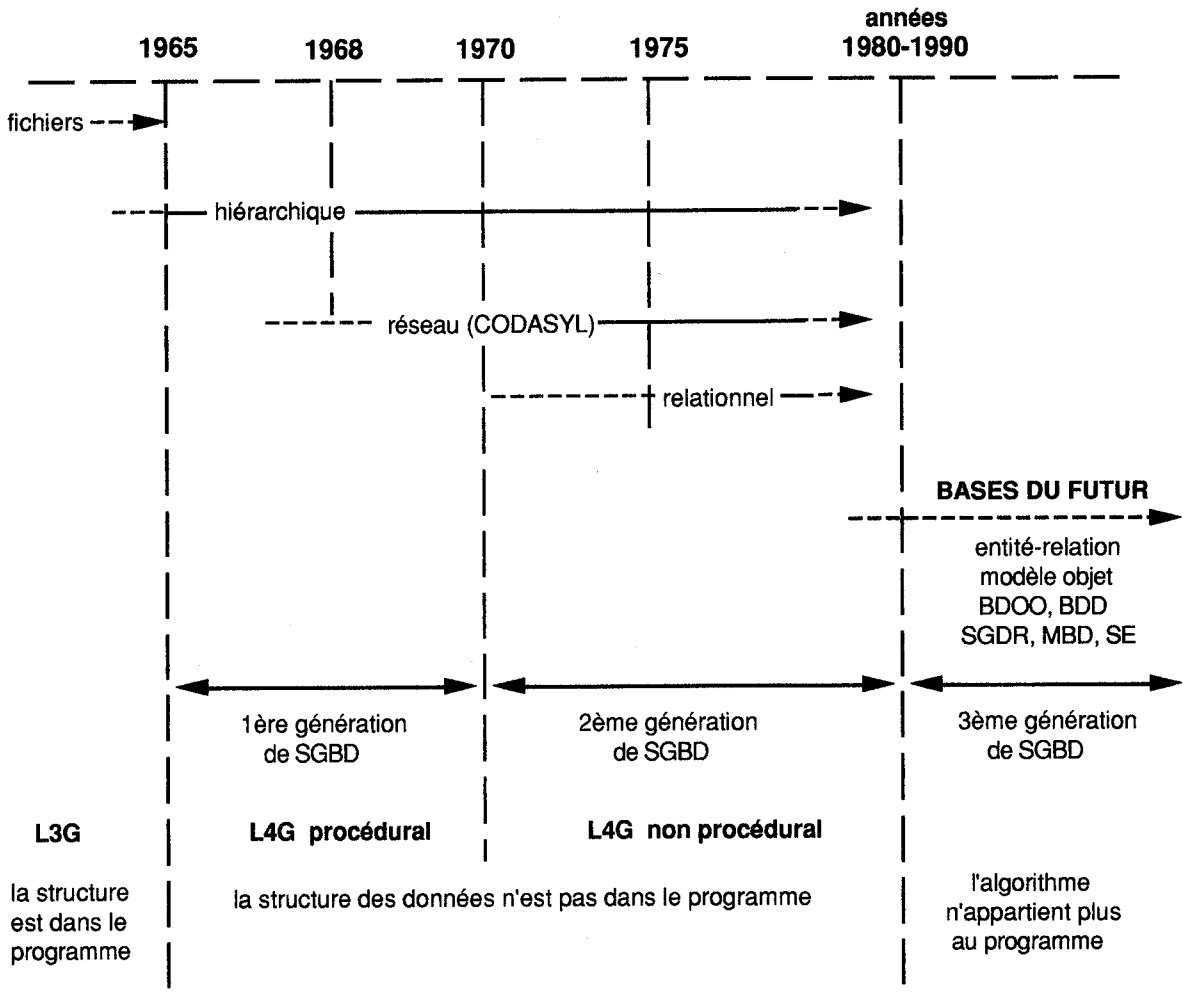


Fig II.2.2.2 : schéma de l'évolution historique des bases de données

II.3 LES RESEAUX INFORMATIQUES (téléinformatiques)

Le sujet de notre étude nous invite à préciser la notion de "réseau". Dans le paragraphe suivant nous donnons une définition et une illustration de son utilisation en constante croissance. Nous abordons aussi le besoin de normalisation. Nous terminons en précisant la position de Digital en matière de réseau, car ces deux éléments sont d'importance dans notre désir de nous préoccuper de la sécurité des SGBD dans cet environnement (cf III.3.1.1.1).

II.3.1 PRESENTATION GENERALE ET EVOLUTION

Les premiers ordinateurs furent implantés en site unique, où la saisie et la production des résultats s'effectuaient au même endroit. Une première évolution est apparue avec les terminaux connectés à distance. Ainsi, se sont développées la saisie décentralisée et la soumission de travaux à distance. Ces systèmes, en se généralisant, ont permis aux utilisateurs de travailler directement par le biais de terminaux sur une base de données centrale.

[GALACSI89] nous indique qu'<<avec les terminaux à distance est apparue la téléinformatique qui a dû résoudre les problèmes de connexion à l'ordinateur central ainsi que la fiabilité des liaisons>>. Il nous précise aussi l'évolution: <<au delà de la simple connexion entre terminaux à distance et un ordinateur, on est passé à la réalisation de la connexion entre deux ordinateurs, puis à celle de réseau téléinformatique (...)

La notion de réseau téléinformatique peut se caractériser comme un ensemble de moyens (lignes de communication, noeuds de communication et d'attente, matériel informatique et logiciel de gestion de réseau), permettant une connexion permanente ou temporaire et un dialogue par échanges d'informations entre ordinateurs à distance>>.

[SIL89] nous confirme cette tendance : <<les quinze dernières années ont été marquées par une rapide évolution dans le domaine de la communication entre ordinateurs. D'abord, sont nés les réseaux à grande distance, permettant de relier entre eux les gros centres informatiques. Puis sont apparus les réseaux locaux. Après des débuts restreints, les réseaux locaux ont atteint un stade de grande diffusion.>>

Cette évolution a bénéficié de l'apport de nouvelles technologies dans le domaine des moyens de transmission comme la fibre optique ou les satellites, mais aussi des progrès en matière de circuit intégré pour l'implémentation des protocoles de communication.

De plus, cette manière de communiquer correspond à un besoin croissant exprimé par les utilisateurs. Ils apprécient les stations de travail: machines quasi-personnelles, de faible coût, au très bonnes performances et surtout très conviviales (souris, écran graphique, multi-fenêtrage...). L'utilisation de ces stations de travail amène très souvent leur intégration dans un réseau local, qui leur permet de partager certaines ressources "coûteuses" comme les disques, les imprimantes laser, les dérouleurs de bande ...

Donc, sans nul doute, cet essor des réseaux va se poursuivre et s'amplifier.

II.3.2 LA NORMALISATION DANS LES RESEAUX

Pour communiquer, les machines doivent obéir à un ensemble de règles (utilisation de langages communs exprimés sous forme de protocoles). Face aux problèmes insolubles de connexions de systèmes trop hétérogènes, la nécessité d'une normalisation s'est imposée.

Aussi, nous pouvons noter que chaque grand constructeur utilise des protocoles spécifiques qui sont par exemple SNA d'IBM, DNA de DEC, DSA de BULL... accompagnés de "passerelles" pour relier les différents mondes.

La normalisation a pour but d'assurer une cohérence d'ensemble au niveau de ces règles (protocole), en fonction des impératifs des différents partenaires impliqués, à savoir les utilisateurs, les constructeurs et les organismes de télécommunication.

Les travaux de normalisation ont débuté dans les années 70, mais les premiers résultats sont apparus dix ans plus tard. Trois organismes ont participé principalement à cette normalisation:

- l'ISO (International Standard Organization) qui regroupe les organismes nationaux de normalisation
- Le CCITT (Comité Consultatif International pour le Télégraphe et le Téléphone) qui regroupe les administrations des télécommunications des pays membres ainsi que les organismes privés officiellement reconnus
- l'ECMA (European Computer Manufacturers Association) qui regroupe les grands constructeurs européens de matériel informatique comme IBM Europe, BULL, ICL, Olivetti ...

C'est l'ISO, dans son modèle de référence OSI (Open System Interconnection), qui s'est attachée à élaborer les normes en matière d'interconnexions de systèmes informatiques.

[PUJO85] nous indique que ce modèle est formé de sept couches successives, chacune d'elle utilisant, pour réaliser ses propres fonctions, les fonctions offertes par la couche inférieure. Les couches du modèle sont numérotées de 1 à 7, cet ordre reprenant un niveau d'abstraction de plus en plus élevé.

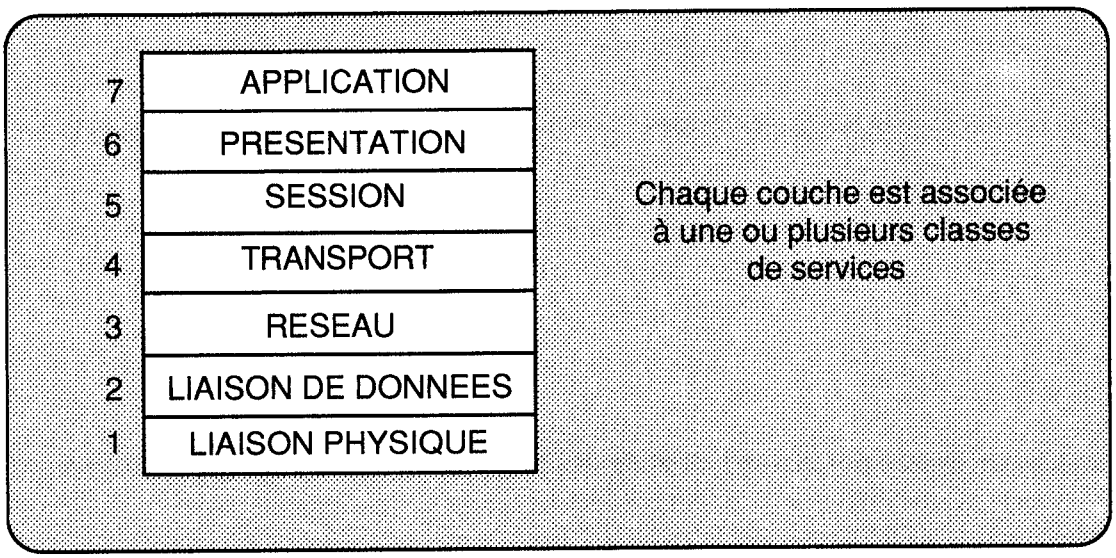


Fig II.3.2 : les sept couches OSI

II.3.3 DIGITAL, LEADER MONDIAL DES RESEAUX INFORMATIQUES

Il nous semble opportun maintenant de préciser la position de Digital en matière de réseau. L'utilisateur de réseau informatique bénéficiera toujours davantage des résultats des travaux en cours, que ce soit de normalisation ou de recherche. Digital s'implique dans ces deux directions :

- *normalisation* en premier lieu, comme l'affirme [DIGITALa] <<OSI est une liberté ouverte sur le monde, à mesure que chaque protocole d'ISO est défini, Digital intègre les spécifications correspondantes. C'est ainsi que Digital a pu proposer dès 1985 des produits conformes aux protocoles OSI avant d'annoncer, en septembre 1987, que la phase V de DNA débouchera sur une conformité totale au modèle OSI>>
- *recherches et évolutions*: Digital s'investit dans le domaine des réseaux. Il s'oriente vers des systèmes en grappes d'ordinateurs (cluster, lavc) mais aussi vers des systèmes qui correspondent bien aux recherches menées en matière de système d'exploitation distribué. Ces systèmes utilisent pleinement les possiblités de communication (cf III.3.1.4 et IV.2) comme l'indique [DIGITALb] : <<VMS 5.4 met également en oeuvre un élément essentiel du traitement transactionnel distribué...>>.

Grâce à ses efforts dans ce domaine, nous pouvons affirmer que Digital se présente aujourd'hui comme le leader mondial des réseaux informatiques [DIGITALa], [DIGITALc].

II.4 LA SECURITE DES SGBD

Après avoir défini les notions de base de données, de SGBD et de réseau, il nous reste à présenter la dernière notion qui apparaît dans le titre de notre sujet: "la sécurité du SGBD en environnement réseau".

La première partie de ce paragraphe donne une définition de la sécurité des SGBD. On verra son extrême importance à travers la multitude des définitions existantes dans la littérature. Nous tenterons de préciser les différents types de sécurité et de mieux cerner le domaine d'investigation qui nous préoccupe:

II.4.1 DEFINITION DE LA SECURITE, LES DIFFERENTES SECURITES

La sécurité apparaît comme une des fonctions que doit assurer un SGBD (cf II.1). On peut rappeler que, pour [ADIB83], elle représente quatre des six fonctions d'un SGBD, pour [MIRA86] un des quatre objectifs et enfin pour [AKOKA84] deux des huit fonctions. Pour [MIRA90] : <<La sécurité des données représente un des problèmes inhérents à l'approche des bases de données qui vise à définir un réservoir commun de données entre tous les utilisateurs>>. De plus, beaucoup d'auteurs y consacrent un, voire même plusieurs paragraphes; elle apparaît donc comme un élément important d'un SGBD.

On peut donner en premier lieu une définition de la notion de sécurité. La littérature étant fournie en la matière, il convient de fixer le vocabulaire:

* pour [MIRA77] la sécurité des données est un terme générique qui comprend trois types de contrôle:
  <<- le contrôle d'accès au système par les utilisateurs non-identifiés ou non authentifiés
  - le contrôle de l'accès illégal aux données, ou confidentialité,
  - le contrôle de la modification invalide des données, ou intégrité>>.

* [MIRA86] revient sur cette proposition et nous soumet deux types de sécurité :
  <<- l'intégrité ou protection contre l'accès invalide aux données (erreurs ou pannes), et contre l'incohérence des données vis-à-vis des contraintes modélisées
  - la confidentialité ou protection contre la modification illégale des données.>>

* [GALACSI89] lui aussi différencie deux notions:
  << - la sécurité qui consiste à conserver intactes les valeurs saisies, quelles que soient les fausses manoeuvres effectuées
  - la confidentialité qui consiste, elle, à ne permettre à un utilisateur qu'à ne connaître et à ne manipuler que les seules valeurs des rubriques dont il a besoin.>>
  On remarque que pour [GALACSI89] la "sécurité-intégrité" de [MIRA86] s'intitule uniquement "sécurité".

* [AKOKA84] propose deux notions, mais le vocabulaire utilisé diffère de celui de [MIRA86]; ainsi parle-t-il de :
  - "sécurité" pour la protection de la base de données contre les accès et modifications illégales non autorisées
  - et d' "intégrité de la base" pour le terme qui, selon lui, a évolué et désigne <<les qualités de validité, de cohérence et l'exactitude des données de la base>>.

* [GARD83] distingue surtout deux idées importantes:
  - <<la résistance aux pannes, c'est le maintien de l'intégrité des données en face des pannes machines
  - la sécurité des données, c'est-à-dire le contrôle des autorisations d'accès aux données et le maintien une certaine confidentialité des données>>.

  Mais il note aussi d'autres aspects de l'intégrité des données:
  - les contrôles d'intégrité
  - le cryptage des données

* [MIRA90] précise les distinctions avancées par [MIRA86] qui déjà différenciait l'intégrité de la confidentialité. Dans son dernier ouvrage [MIRA90] mentionne deux types d'intégrité complémentaires:
  - l'intégrité interne qui concerne la sémantique des données de la base (contrainte du schéma, contrainte propre au monde réel)
  - l'intégrité externe qui concerne le contrôle de la concurrence entre plusieurs utilisateurs et la reprise sur panne.

* On peut terminer cette longue liste de définitions par celle d'[ADIB83]. Il distingue quatres notions assurant la sécurité de la base et gérant les problèmes d'intégrité. Elles correspondent à quatre des six fonctions du SGBD (cf II.1):
  - l'intégrité interne qui garantit la cohérence des informations stockées par rapport à leur signification (ex: l'âge < 120)
  - la synchronisation des accès concurrents qui autorise plusieurs usagers à manipuler de manière concurrente la même base de données, en garantissant que les actions des uns ne viennent porter préjudice aux actions des autres
  - la sécurité d'utilisation qui contrôle la manipulation de la base par les seuls usagers autorisés
  - la sûreté de fonctionnement qui permet au SGBD, après une panne pouvant altérer les données de la base, de retrouver un état cohérent grâce à l'utilisation d'un mécanisme de reprise.

II.4.2 LA SECURITE AU SENS "INTEGRITE DES DONNEES" (sûreté de fonctionnement, traitement des pannes)

II.4.2.1 La sécurité (sûreté) de fonctionnement

Nous avons vu au paragraphe précédent que la littérature est fournie en matière de définitions sur la sécurité. Cela nous a permis de préciser les différents types de sécurité (niveau). Nous pouvons maintenant mieux cerner le domaine d'investigation de notre étude, en indiquant au travers de ces définitions la façon dont s'intitulerait la sécurité qui nous préoccupe:

* "intégrité, reprise sur panne" pour [MIRA90],
* "sécurité" pour [GALACSI89],
* "l'intégrité des données, résistance sur panne" pour [GARD83],
* "intégrité" pour [AKOKA84],
* "sûreté de fonctionnement, mécanisme de reprise" pour [ADIB83].

Nous pouvons donc indiquer que nous allons traiter de la "sécurité" en tant que "sûreté de fonctionnement", activée consécutivement à un incident, pour assurer "l'intégrité externe" des données, et ce, grâce à un "mécanisme de reprise sur panne".

II.4.2.2 Les différents types de pannes

[ADIB83] nous en propose trois principaux:
- "la défaillance machine" suite à une panne de courant ou à une défaillance matérielle : la machine s'arrête brutalement alors que le SGBD est actif. En général, ce genre de panne a pour effet de détruire le contenu de la mémoire volatile (mémoire centrale), mais le contenu des disques et bandes est préservé
- "la défaillance d'un périphérique" correspond par exemple à une destruction totale ou partielle du contenu d'un disque (écrasement des têtes, poussières...)
- "la défaillance logicielle" survient pendant l'exécution du SGBD; celui-ci peut être interrompu par une erreur logicielle nécessitant un redémarrage, provoquant une perte de la mémoire volatile, et laissant les disques dans un état plus ou moins cohérent.

Il note aussi que des causes plus graves peuvent entraîner une destruction totale ou partielle des informations et du matériel : "feu, attentat, inondation..."

[MIRA90], quant à lui, nous indique que les pannes dépendent du type de mémoire. Il en distingue trois :
- volatile (exemple: mémoire centrale)
- en ligne non-volatile (exemple: disque)
- hors ligne non-volatile (exemple: bandes d'archives)

[GARD83] expose quatre types de panne:
- "la panne d'une action" survient quand une commande du SGBD est mal exécutée
- "la panne d'une transaction" survient à la suite d'une erreur de programmation, d'un verrou mortel, d'une panne d'action non corrigible
- "la panne du système" provoque l'arrêt du système et son redémarrage; la mémoire centrale est perdue
- "la panne de mémoire secondaire" surgit soit après une défaillance matérielle, soit à la suite d'une défaillance logicielle impliquant une mauvaise écriture. Dans ce cas là, une partie de la mémoire secondaire est perdue. C'est la panne la plus catastrophique.

Il nous indique, en plus, la fréquence de ces pannes : les deux premiers types de panne peuvent apparaître plusieurs fois par minute alors que les pannes systèmes surviennent généralement plusieurs fois par mois. Une panne secondaire, elle, ne se produit qu'annuellement.

Remarque: on notera que les pannes réseaux ne figurent pas dans ces listes. On reviendra sur ce point dans le paragraphe II.4.3 "sécurité de fonctionnement dans un environnement réseau" afin de positionner ce type d'incident qui nous intéresse.

II.4.2.3 Mécanisme de reprise sur panne

Après avoir cerné les différents types de pannes, nous abordons les mécanismes de reprise sur panne.

a) Notion de sauvegarde:

[ADIB83] nous propose une première approche du problème : <<la méthode la plus classique pour résister à toutes ces différentes formes de défaillances ou de pannes, est d'effectuer à intervalles réguliers des copies de l'état de la base (sur bande par exemple) et de stocker ces bandes en lieu sûr. Ainsi, au pire, peut-on revenir à l'état de la base au jour de la dernière sauvegarde>>

Mais il précise tout de même qu'<<une autre classe de méthode pour résister aux pannes consiste à tenir un journal de toutes les opérations effectuées sur la base>>.

On peut maintenant définir les notions de "journal" et de "point de reprise".

b) Notion de journal:

[GALACSI89],[GARD83] distinguent deux types de journaux:
- un journal de type A "after" (après)(after image log) qui archive les nouvelles pages du fichier permanent (support de la base)
- un journal de type B "before" (avant) (quick-rapide) (before image log) qui archive les pages avant modification

Le schéma suivant figure les transferts de chaque page lors d'une modification d'un fichier permanent (support de base).

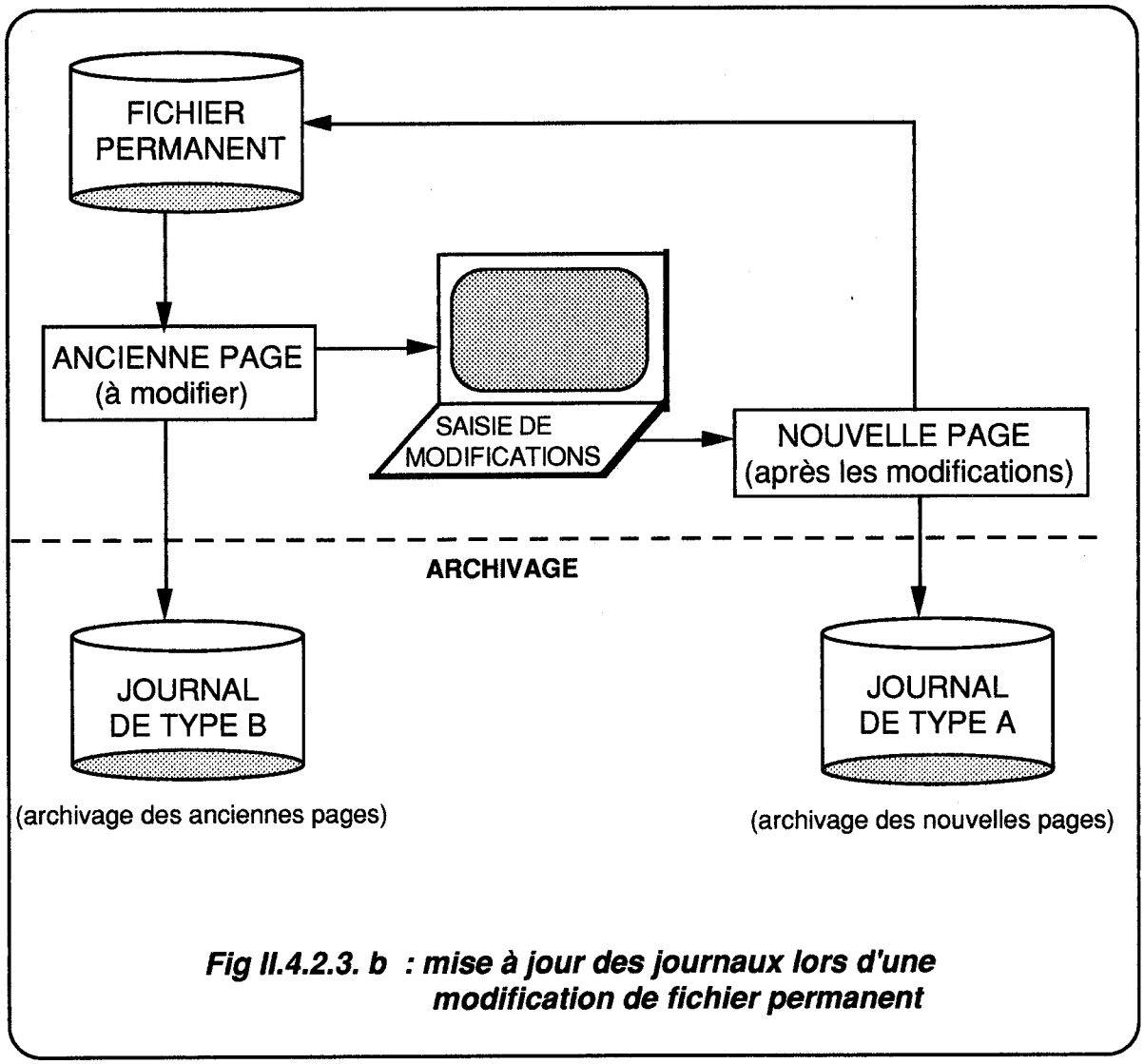


Fig II.4.2.3. b : mise à jour des journaux lors d'une modification de fichier permanent

Les journaux permettent de reconstituer les états successifs d'un fichier permanent. Il suffit d'indiquer l'instant désiré pour effectuer la reconstitution.

c) Notion de point de reprise:

[GALACSI89],[GARD83] : <<les points de reprise ("check-point") sont des instants définis automatiquement à intervalles réguliers. A chaque point de reprise, l'état des divers mémoires et registres est sauvegardé ainsi que les explications nécessaires pour que chaque programme en cours d'exécution puisse être repris à l'endroit même où il était à l'instant du point de reprise>>. [CLIOb] : <<c'est un point de contrôle du système, il sert de point de reprise en cas d'erreur ou d'étreinte fatale. Il s'agit d'un point de cohérence physique de la base. Les "pages" modifiées en mémoire sont écrites sur la base; les ressources sont libérées; le contexte de l'utilisateur est sauvegardé; le journal B est remis à zéro>>.

d) Mécanisme (procédure) de reprise, restauration d'un fichier permanent:

[GALACSI89] : En cas d'incident, le SGBD reconstitue le système d'informations dans l'état où il était au dernier point de reprise. Il y a, en particulier, restauration de chaque fichier permanent. Il existe deux types de restauration [MIRA90], [GARD83], [GALACSI89], [GRAY78]:

- la reprise (restauration) à chaud
- la reprise (restauration) à froid.

La restauration à chaud :

[GALACSI89] : <<cette restauration n'est possible que si l'incident n'a pas détérioré le fichier permanent, il y a seulement une perturbation du fonctionnement normal de l'ordinateur (les usagers restent en ligne)>>, ce que [MIRA90] indique comme: <<"panne concernant la mémoire volatile", une panne d'entrée sortie non recouvrable, une panne légère ne touchant qu'une petite partie de la base de données (interblocage...)>>.

Il faut reconstituer le fichier permanent dans l'état où il se trouvait lors du dernier point de reprise et, pour cela, retrouver les anciennes pages dans le journal de type B, comme l'illustre la figure ci-après.

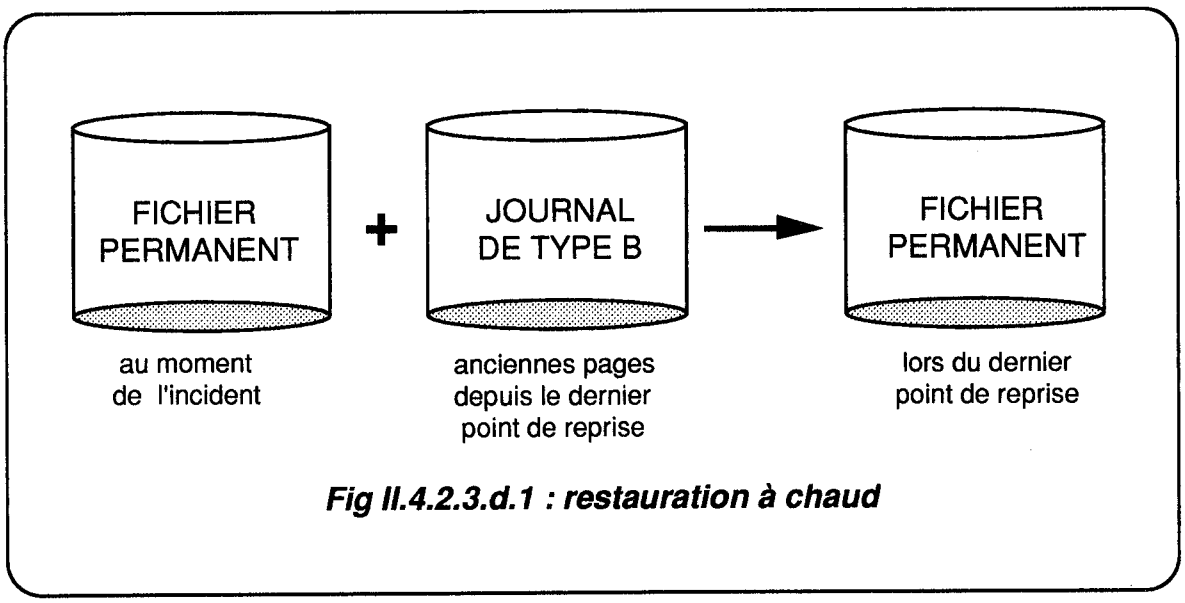


Fig II.4.2.3.d.1 : restauration à chaud

la restauration à froid:

[GALACSI89] : <<le fichier permanent est inutilisable après l'incident (la session conversationnelle en cours est arrêtée). On a recours à la sauvegarde créée en début de période, il faut lui ajouter toutes les nouvelles pages définies depuis, jusqu'au dernier point de reprise. On va les chercher dans le journal de type A qui est remis à zéro après chaque création de sauvegarde>>:

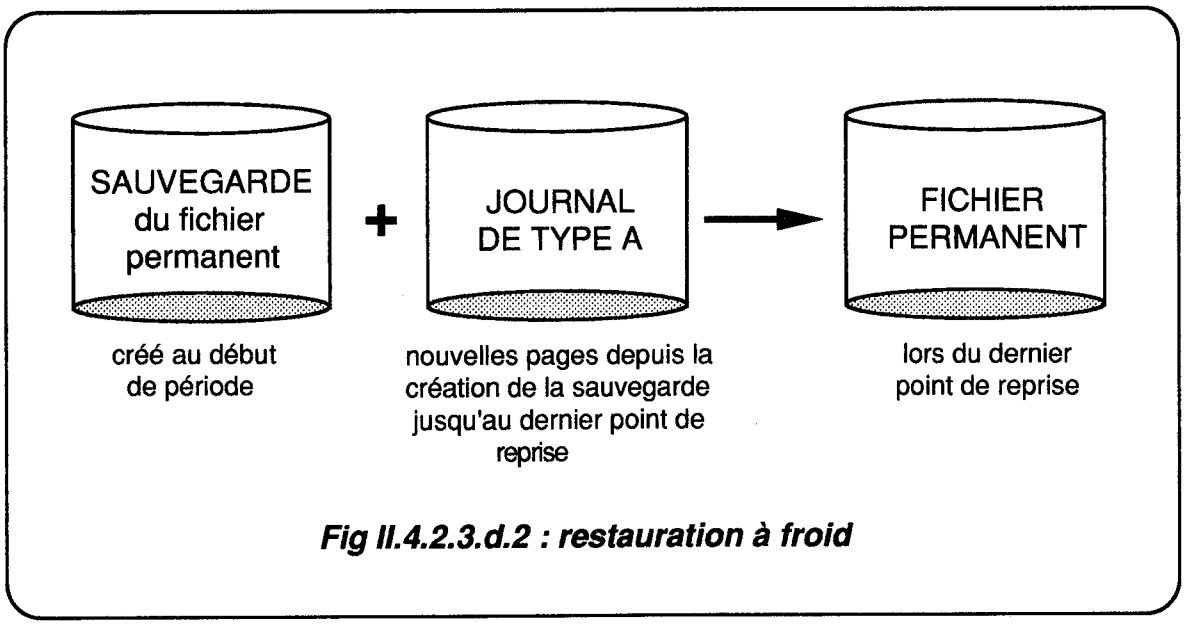


Fig II.4.2.3.d.2 : restauration à froid

Remarque: il existe aussi un autre principe que celui de l'utilisation de journaux: il s'agit de la notion de "page ombre" que nous ne détaillerons pas ici.

II.4.3 LA SECURITE DE FONCTIONNEMENT DANS UN ENVIRONNEMENT RESEAU INFORMATIQUE

Nous avons vu que, pour définir la sécurité qui nous intéresse, les auteurs utilisaient des termes différents: "intégrité", "sûreté de fonctionnement". Nous entendons par "sécurité de fonctionnement dans un environnement réseau informatique", un système qui assure l'intégrité (physique, externe) des données de la base sur toutes les pannes qui peuvent survenir dans les différents éléments du réseau (lignes de communication, noeuds de communication, multiplexeur ...). Il exécute pour cela une procédure de restauration. On considère là la défaillance d'un élément du réseau et non la panne mécanique d'un ordinateur isolé, qui correspondrait à une panne de type unité centrale, disque ou secteur .

Essayons maintenant de noter si des auteurs ont étudié ce type de pannes (cf chapitre III). Nous abordons ici les définitions et les notions théoriques. Il faut relever tout d'abord que des ouvrages comme [GARD83] ou [ADIB83] ne font pas apparaître les pannes réseaux dans leurs listes des pannes possibles.

[COHEN89] et [AKOKA84] dans leurs études comparatives sur les SGBD soulèvent le problème de la sécurité des SGBD mais ils ne traitent pas précisément des incidents réseaux.

De même [GALACSI89] évoque <<les problèmes des perturbations des communications et des pannes qui peuvent survenir même sur un réseau stable>> et il nous indique que <<le système de gestion de base de données doit prévoir les procédures de reprise en cas d'incident et de prévention de la situation incohérente>>.

[PHI86] traite le problème dans un paragraphe sur les facteurs d'incohérence et les procédures de reprise en indiquant que les principales causes d'apparition d'état non cohérent dans une base de données sont les pannes.Il précise à propos des bases de données en environnement réseau défaillant: <<s'il s'agit d'une coupure de ligne, l'état de la base est restitué, on peut par sécurité lancer une procédure de rollback (retour arrière)>>.De plus, il différencie les pannes réseaux des incidents du type disque détruit ou secteur qui nécessitent de disposer respectivement d'un "journal après", et d'une restauration à chaud.

[GARVAL85] nous donne un exemple de base de données qui <<détecte une panne de machine hôte ou de réseau>> et nous indique <<que l'intégrité des données est en principe maintenue>> et que, de plus, <<à partir du journal, la procédure de reprise est exécutée et toutes les mises à jour sont défaites jusqu'au dernier point de reprise>>.

On peut conclure que peu d'auteurs traitent du problème de l'incident réseau mais ils nous permettent de différencier cet incident des autres types de pannes. Ils nous indiquent, de plus, que la sécurité activée semble basée sur l'utilisation d'un journal (comme cela reste le cas dans la sécurité classique).

CHAPITRE III

CLIOPS

Après avoir exposé le projet, son cadre et son environnement, et explicité les notions théoriques introduites par le titre de notre sujet, ce troisième chapitre majeur, décomposé en cinq parties, aborde pleinement notre étude. La première présente de façon synthétique le SGBD CLIO et sa sécurité standard, sécurité que nous cherchons à perfectionner. La deuxième examine notre choix méthodologique de la méthode Merise. La troisième est consacrée à l'étude de notre conception du projet. Les deux dernières parties livrent l'expérience enrichissante constituée par le maquettage et la réalisation de notre projet. Nous soulevons, à cette occasion, les difficultés rencontrées ainsi que les enseignements tirés.

III.1 CLIO

III.1.0 HISTORIQUE DE SOCRATE-CLIO

1969 - 1970 : définition du SGBD à l'Université de GRENOBLE
1970 : industrialisation de SOCRATE sur IRIS 50
1972 : début du portage sur IBM sous DOS
1975 : commercialisation de SOCRATE V1.5
1983 : disponibilité d'une "souche portable" et d'un SOCRATE niveau V1.7 sur IRIS 50, DPS8, DPS7, DPS6, SOLAR, IBM sous OS, DOS et VM-CMS, VAX sous VMS
1984 : CLIO : nouveau produit compatible avec SOCRATE mais dont l'architecture utilise au mieux les systèmes d'exploitation et les nouvelles fonctionnalités, tant pour les informaticiens que pour les non informaticiens.
1985 : CLIO sur micro à base de 68000 (MICROMEGA)
1986 : CLIO sous SPS7; Outil d'interrogation guidée (QUERY)
1987 : prototypeur et générateur d'application CLIO sur Matra Datasystème
1988 : CLIO sur PC
1989 : CLIO V4 (turbo), QUERY PC, CLIO COOPERATIF
1990 : CLIO V6 (SQL)

III.1.1 PRESENTATION GENERALE DE CLIO

CLIO est un logiciel de type SGBD, c'est-à-dire un Système de Gestion de Base de Données (cf II.2). CLIO est apparu en 1984 dans la filiale grenobloise de SYSECA LOGICIEL. Il a pour objectif [CLIOa]:

- l'amélioration de la productivité en développement et en maintenance
- l'aide à la décision pour l'utilisateur final en lui offrant des outils très conviviaux d'accès à une base centrale ou de développement d'une base personnelle
- la sécurité, tant sur le plan de l'exploitation et de l'intégrité des données, que sur le plan du retour d'investissement des coûts de développement.

[CLIOa] "CLIO est un SGBD de type réseau" comme nous avons pu le définir au paragraphe II.2 et comme nous le confirme [MIRA86] dans sa liste de SGBD réseau, ainsi qu' [ADIB83] dans son étude des SGBD. CLIO est un SGBD autour duquel gravitent de nombreux outils d'aide au développement tels que: Atelier de génie logiciel, Editeurs d'états, Gestionnaires d'écrans, L4G, Interface langage, Prototypeur, Query:

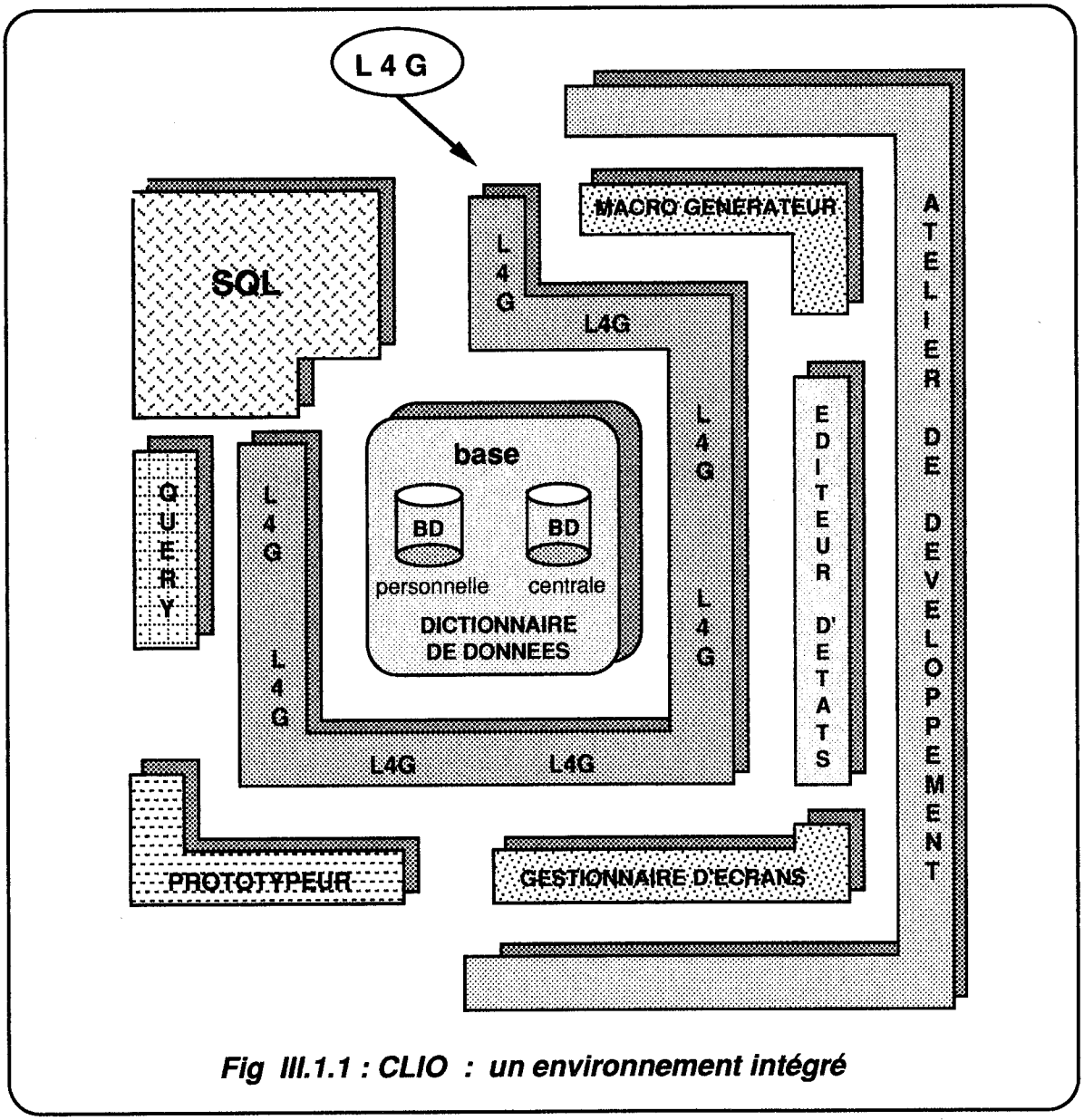


Fig III.1.1 : CLIO : un environnement intégré

Une des particularités de CLIO, est d'être présent sur un parc très diversifié de matériels. En effet, CLIO est un produit portable qui fonctionne sur les machines suivantes:

--> IBM : série 9370,43xx, 30xx sous DOS, MVS, CICS et TSO
--> BULL : DPS8 sous GCOS8, TDS, TSS, DPS7 sous GCOS7, TDS, IOF, DPS6 sous GCOS6 SPS7 sous unix V
--> DEC : VAX et MICROVAX sous VMS
--> HP : HP 9000 sous unix
--> SMH-ALCATEL : MICROMEGA sous MIMOS
--> MATRA DATASYSTEME : série MS1300, MD500 sous unix
--> PC et compatible sous MS-DOS environnement MS-WINDOWS.

Pour tous ces aspects, CLIO représente un produit performant, installé sur de nombreux sites (plus de 900 à ce jour).

III.1.2 LA SECURITE STANDARD DE CLIO:

La sécurité CLIO a les mêmes objectifs que ceux définis dans la partie théorique de notre étude sur la sécurité des SGBD (cf II.4):

* la sécurité directe d'utilisation qui se compose de:
  - la sûreté de fonctionnement (journalisation et reprise sur panne)
  - l'intégrité sémantique (interne) des données
  - la gestion des accès concurrents
* la confidentialité d'accès aux données
* la pérennité de CLIO.

Dans ce paragraphe, nous ne parlerons que de la sécurité qui nous préoccupe, c'est-à-dire la sûreté de fonctionnement. Elle est basée sur les mêmes principes que ceux présentés au chapitre II, à savoir:

- sauvegarde
- journalisation
- point de contrôle
- mécanisme de restauration.

a) Sauvegarde CLIO:

Il est possible d'effectuer des sauvegardes d'un ou plusieurs espaces de la base, ceci grâce à un utilitaire CLIO. Nous pouvons aussi, sans arrêt des utilisateurs, procéder à une sauvegarde globale de la base (sauvegarde en ligne).

b) Journaux CLIO:

Le système CLIO gère deux journaux :

* le "journal rapide" (Journal Quick ou JQ) dans lequel sont notées les images des pages de la base avant la mise à jour. Ce journal permet les "reprises à chaud" - journaux de type B (before) (cf II.4)
* le "journal de reprise à froid" qui inscrit les images des pages modifiées après la mise à jour - journaux de type A (after) (cf II.4)

c) Point de reprise CLIO:

Avec CLIO nous définissons deux types de point de reprise:

* point de référence : c'est le point de départ du dispositif de sécurité. Il doit correspondre à un état de la base tel que :
  - la base est physiquement cohérente
  - la base est sémantiquement cohérente
  - il existe une sauvegarde.

  La prise du point de référence est à la charge du responsable de la base.

* point de contrôle : plusieurs connexions sur une base peuvent se dérouler entre deux points de référence. Au cours de ces connexions, la base peut avoir été modifiée.
  Ces connexions sont découpées par l'intermédiaire de points de contrôle (check-point ou points de reprise, cf II.4).
  Un point de contrôle correspond à :
  - une base physiquement cohérente
  - un état dans lequel la base pourra éventuellement être ramenée par les procédures de restauration (automatique ou différée). Ceci signifie, entre autre, que l'utilisateur est capable de reprendre le traitement en cours à partir de ce point de contrôle.

  De plus, en un point de contrôle, CLIO assure:
  - l'écoulement des entrées/sorties en cours d'exécution par le système d'exploitation
  - la recopie sur la base des informations modifiées en mémoire centrale
  - la libération des ressources allouées à l'utilisateur
  - le marquage du fichier journal Q (rapide).

  Il est à noter qu'un point de contrôle concerne un utilisateur, indépendamment des points de contrôle des autres utilisateurs éventuels de la base.

  La prise du point de contrôle s'effectue de deux façons:
  - une prise implicite de point de contrôle:
    le système CLIO prend implicitement un certain nombre de points de contrôle:
    * lors d'une demande de connexion ou de déconnexion
    * en début et fin de transaction
    * dans une transaction utilisant le gestionnaire d'écran, avant et après chaque appel à une macro du gestionnaire d'écran
    * après l'exécution de commandes (1er niveau) et de certaines fonctions.
  - une prise explicite de point de contrôle: au cours de l'exécution d'une transaction, on peut demander la prise d'un point de contrôle par la requête CKPT.

d) Mise en oeuvre de la restauration CLIO :

Nous n'abordons pas ici la restauration à froid, qui correspond à une restauration classique (comme nous l'avons définie au paragraphe II.4), mais nous nous intéressons uniquement au fonctionnement et vocabulaire de celle qui nous préoccupe : la sécurité mettant en oeuvre les journaux rapides (JNLQ ou JQ).
Avant de l'expliciter, nous pouvons revenir sur la notion de Journal Quick .

d.1) Journal Quick (Journal Q, JNLQ ou JQ):

Le fichier de journalisation rapide ou journal Q est utilisé autant pour les reprises automatiques que pour les reprises différées. Les informations notées au niveau du journal Q permettent de ramener la base à l'état du dernier point de contrôle, en partant de l'état final de la base. Ce journal est enregistré sur disque. On rappelle qu'il permet de stocker, entre deux points de contrôle, toutes les pages faisant l'objet d'une mise à jour, avant leur modification. En cas d'incident, le journal

Q permet de restaurer la base au point de contrôle précédent, en remplaçant les pages modifiées par leur valeur avant modification.

De plus, il existe un journal Q par utilisateur de CLIO. Ce fichier, créé (ou alloué) en début de connexion, est invalidé à la fin de celle-ci.

d.2) Restauration d'une base

Soient B0 l'état de la base au début de la session, cet état correspondant à un point de référence et Bi l'état de la base à l'instant du point de contrôle i (appelé aussi check-point):

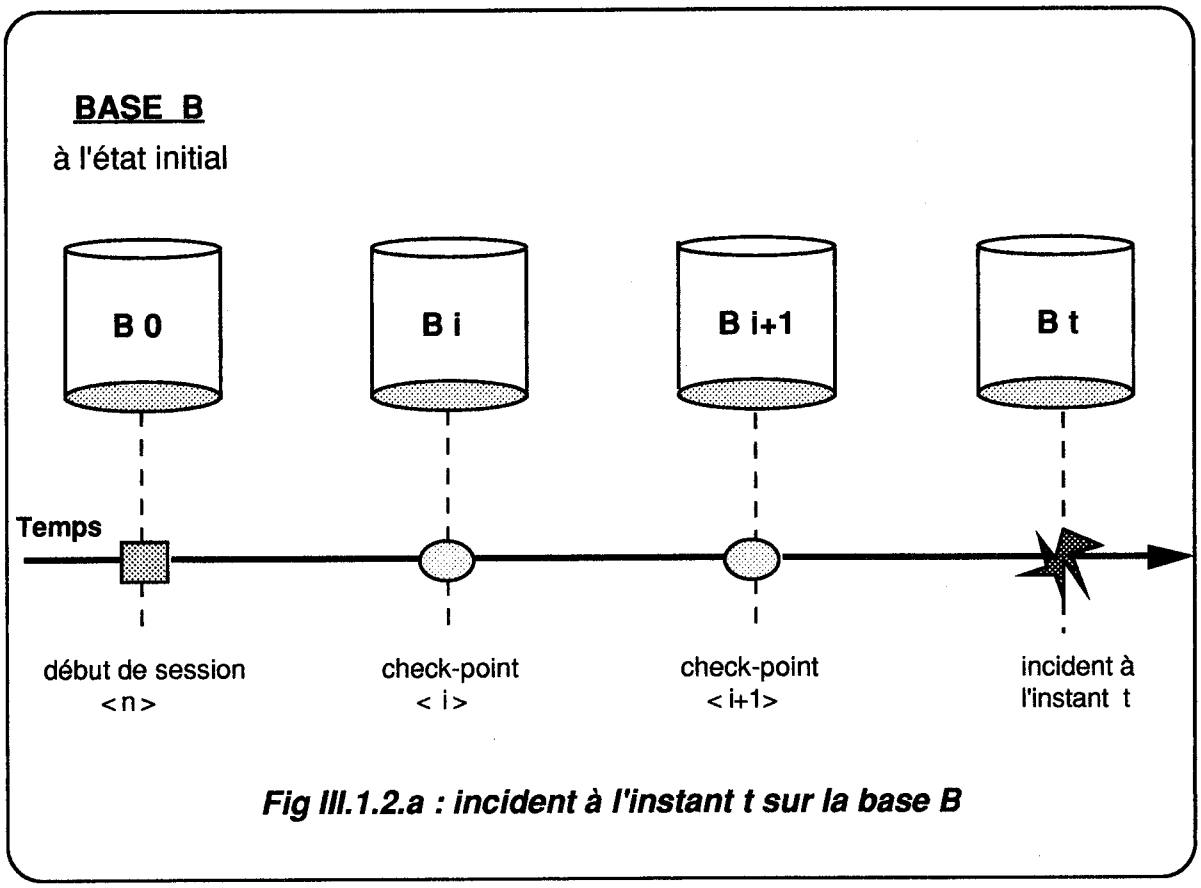


Fig III.1.2.a : incident à l'instant t sur la base B

Si nous supposons qu'un incident est survenu à l'instant "t", nous allons étudier quelle est la possibilité de restauration dont dispose l'utilisateur pour ramener sa base à l'état cohérent Bi+1.

Si le support physique de la base est détruit, nous n'avons pas de restauration possible à partir du journal Q: il faut repartir d'une restauration à froid (cf II.4). Par contre, si l'état final Bt peut être utilisé, on peut ramener la base à l'état Bi+1 correspondant au dernier point de contrôle effectué, comme l'illustre le schéma suivant:

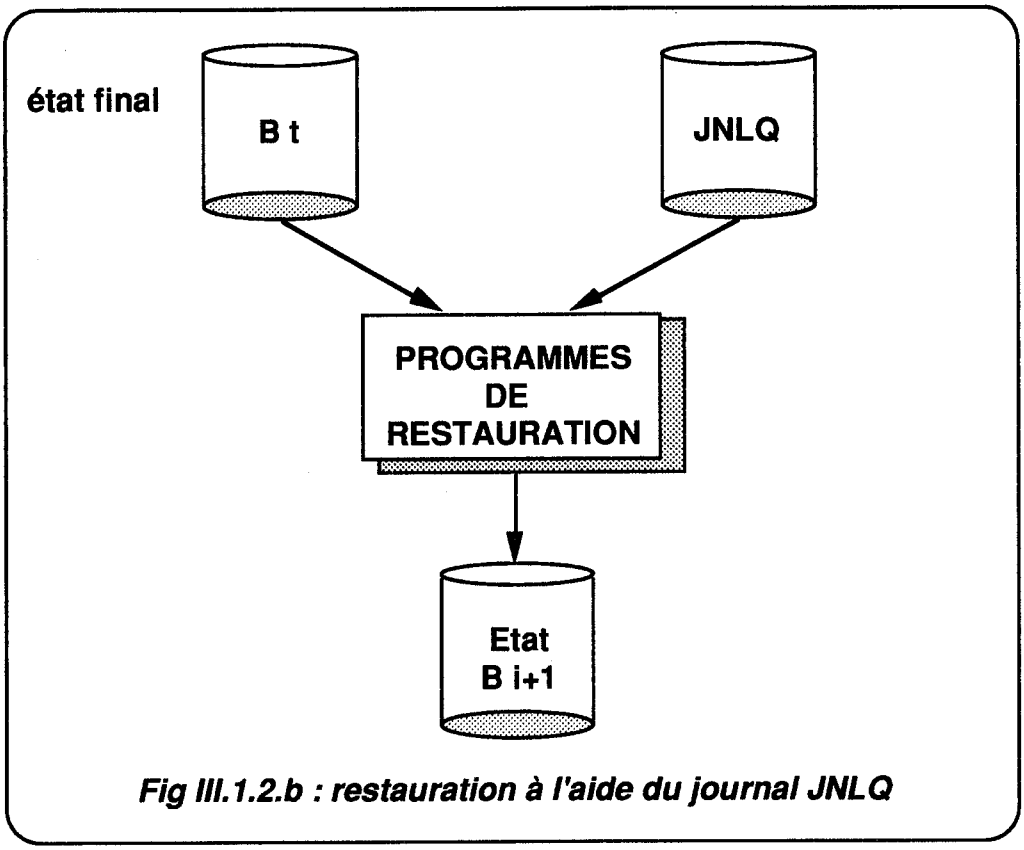


Fig III.1.2.b : restauration à l'aide du journal JNLQ

Deux méthodes sont utilisables pour effectuer cette restauration; leur emploi est déterminé par le type d'incident:

Restauration automatique:

La restauration automatique de la base est activée :

* lors d'une erreur CLIO de niveau sévère, terminal ou ultime (provoquant en plus l'arrêt de l'utilisateur)
* lors de l'arrêt de CLIO causé par un problème logiciel
* lors d'une erreur utilisateur (arrêt par CTRL Y, erreur du module système entraînant une erreur VMS et l'arrêt de CLIO).

La restauration est réalisée par un Exit Handler, automatiquement activé. (cf III.3). Après la restauration, le journal Q est invalidé.

Restauration différée:

La restauration différée sera employée consécutivement à un arrêt de la machine (problème matériel ou coupure de courant). Elle s'effectuera donc en utilisant tous les journaux Q existants. Elle est activée au redémarrage de la machine.

On peut conclure notre présentation de la sécurité standard de CLIO, en insistant sur le fait que la sécurité CLIO, assurant l'intégrité physique des données, repose sur la restauration d'un journal Q, restauration qui doit s'effectuer au moment opportun. (Ce point est important car il s'agit du principe de base réutilisé dans l'adjonction de sécurité "CLIOPS".)

III.2 LA METHODOLOGIE UTILISEE POUR CLIOPS :

Au départ de notre étude, nous avons réfléchi à l'opportunité d'employer une méthode de conception et dans l'affirmative, il a fallu la déterminer. C'est cette réflexion et ce choix que nous vous livrons dans ce paragraphe ainsi qu'une présentation de la méthode et de la répartition du travail de l'équipe.

III.2.0 POURQUOI UNE METHODE ?

Le cours du soir de Méthodologie d'Informatisation nous avait déjà fait ressortir le bien fondé de l'utilisation d'une méthode de conception. Nous pouvons reprendre là les arguments classiques que l'on nous a enseignés et qui nous ont convaincus:

- éviter l'empirisme individuel et permettre une coopération efficace entre les différentes personnes impliquées
- permettre la communication entre individus de l'équipe de conception
- construire des systèmes pertinents, fiables, flexibles et adaptatifs
- permettre d'évaluer le système à tout moment de son cycle
- améliorer les coûts et délais de production.

De plus, le projet "CLIOPS" comportait une phase d'analyse conséquente et nécessitait aussi une communication d'informations non négligeable entre concepteur et superviseur d'étude interne. J'avais donc à coeur de m'entourer d'une méthode guidant ma démarche et jalonnant les diverses étapes de mon travail. La méthode devait, autant que faire se peut, m'aider à produire des documents permettant le contrôle, la validation et la communication ...

III.2.1 APPROCHE SELON LA METHODE MERISE :

III.2.1.0 Pourquoi Merise?

SYSECA emploie habituellement différentes méthodes, et plus particulièrement une méthode maison "MEDOC" lorsqu'il s'agit de grands projets (ex : MUST). D'autre part, j'avais, par le cours de Méthodologie d'Informatisation du CNAM, eu connaissance d'un certain nombre de méthodes. Je décidais donc d'effectuer mon choix parmi celles enseignées au CNAM et celles employées par SYSECA.

Durant mon cours de Méthodologie, nous avions eu une sensibilisation à SADT, AXIAL , IDA et un apprentissage plus conséquent de Merise, approfondi par des Travaux Pratiques d'une année. Je n'avais, par contre, aucune connaissance de MEDOC. Je décidais donc d'effectuer une petite étude comparative, dont je livre ici le tableau récapitulatif:

	MERISE	SADT, AXIAL, IDA	MEDOC
Connaissance de la méthode	connu	sensibilisation	inconnu
Connaissance de la méthode par les autres membres de l'équipe	certains	certains	aucun
Formalisme	simple,naturel	moins simple	divers
Souplesse d'utilisation	souple	moins souple	moins souple
Emission de document	oui	oui	oui
Adapté au projet	s'adapte	SADT : oui	pour plus gros projets

Fig III.2.1.0 : étude des différentes méthodologies

L'ampleur du projet ne me permettant pas d'une part, de consacrer beaucoup de temps à l'apprentissage d'une nouvelle méthode et d'autre part, d'en employer une trop lourde, je décidais donc d'adopter Merise.

Cette méthode possédait les avantages de m'être familière, de présenter un formalisme clair et connu par une partie de l'équipe, ainsi qu'une souplesse d'utilisation rendant possible un emploi dans sa non-intégralité.

De plus, elle répondait, tout au long de la conception, à la nécessité de produire des documents permettant un suivi par le responsable des études internes, mais constituant surtout un élément de validation de chaque étape. Il m'apparut alors que le choix de Merise s'inscrivait dans un bon compromis.

III.2.1.1 La méthode Merise:

<<Merise est une méthode de conception et de développement de systèmes d'informations>> [TAROCO86]. On peut illustrer ses origines par le schéma suivant:

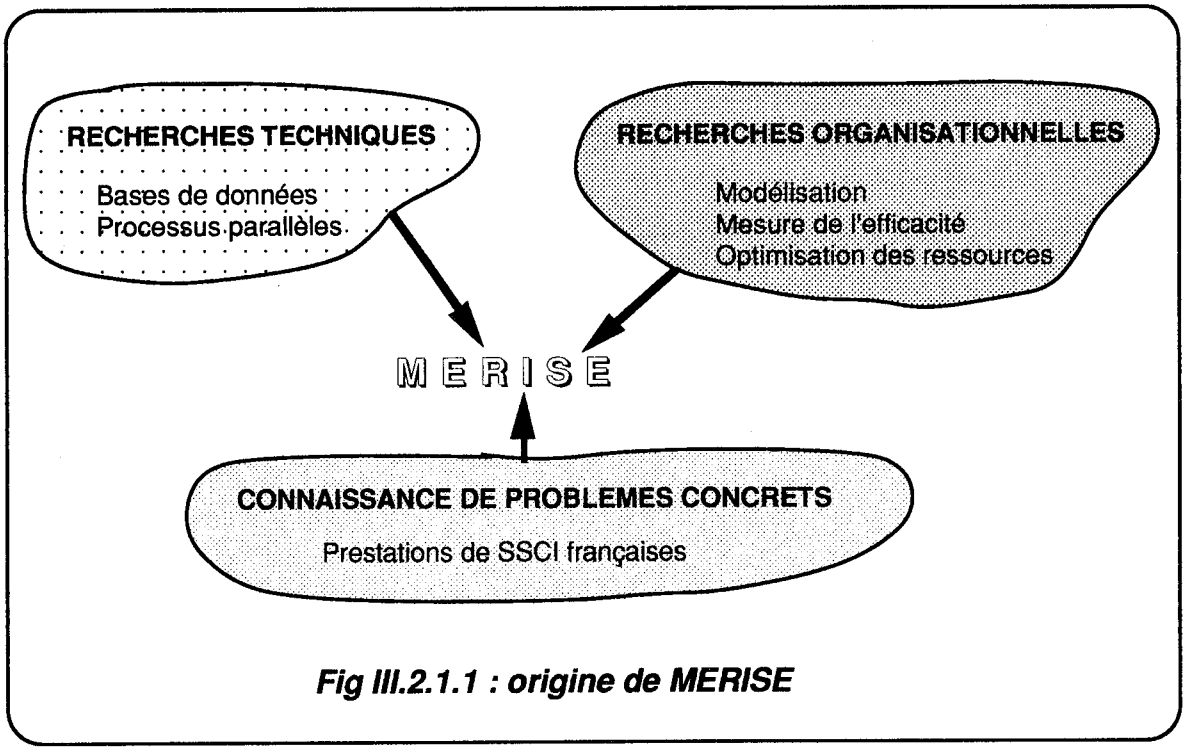


Fig III.2.1.1 : origine de MERISE

Dès 1976, le Ministère de l'Industrie prit conscience de l'importance des méthodes. Il en découla que la "Mission informatique" lança l'étude et la publication des Guides Racines (établissement d'un schéma directeur) et Actif.

C'est dans ce contexte et durant les années 1978-1979, que la méthode Merise fut développée, mise au point et expérimentée par le CETE d'Aix en Provence ainsi que par plusieurs Sociétés de Service et de Conseil en Informatique. Il s'agissait de répondre à l'attente des utilisateurs, conscients de l'obsolescence des méthodes de première génération (Cartésiennes).

Merise constitue avant tout une philosophie, caractérisée par une approche globale qui permet une analyse du système par niveau de préoccupation, ainsi qu'un dialogue et une validation permanente. Cette méthode globale d'analyse (systémique) se distingue aussi par sa démarche rigoureuse, ses modèles, ses normes, et son formalisme.

La démarche de Merise cherche à définir :
- les étapes et les phases qui jalonnent la vie du Système d'Informations
- les prérequis et les résultats attendus pour chaque étape
- les décisions portant sur les objectifs et la planification des moyens.

Le formalisme de Merise est hérité, entre autre, de la théorie des ensembles et des réseaux de Pétri.

[01INFO88a] définit Merise comme <<une méthode complète. Le niveau de représentation conceptuel, tant des données que des traitements, permet de répondre au "QUOI". Le modèle logique par delà le changement de formalisme permet la réponse au "QUI FAIT QUOI ET OU". Le niveau physique, enfin s'intéresse au "COMMENT">>.

III.2.1.2 Utilisation de la méthode et répartition du travail au sein de l'équipe:

Suite à ma proposition d'évolution, j'ai été amené à prendre en charge la conception du projet.

Par respect de la méthodologie utilisée, et par nécessité de suivre l'état d'avancement de l'étude, je me suis engagé à produire des documents rendant compte de l'analyse effectuée.(cf III.3)

Avec le même souci, et afin de pratiquer la validation de l'étude, je soumettais mon travail, après chaque étape clé, aux autres membres de l'équipe (discussions à partir de documents, souvent source de polémiques édifiantes). Ces réunions m'obligeaient parfois à reprendre mon travail mais faisaient toujours progresser le projet.

Les "experts" SGBD de SYSECA ont été eux aussi sollicités, durant la première partie de la phase de conception, afin d'obtenir certains conseils et de profiter de leur grande expérience.

Durant la dernière étape du projet (réalisation et intégration de "CLIOPS" cf III.5), stagiaires et membres de l'équipe furent de l'ouvrage.

Les différentes relations du concepteur avec le demandeur, le valideur ou le réalisateur seront livrées aux paragraphes "aspect relationnel du projet" des chapitres suivants.

Nous pouvons noter que notre préoccupation fut de toujours pratiquer une phase de validation après chaque avancement du projet, afin de progresser en toute sécurité et de ne pas nous engager dans une impasse ou une mauvaise direction.

III.3 CONCEPTION DU PROJET CLIOPS

Dans le paragraphe précédent, nous avons justifié et présenté notre choix méthodologique utilisé pour la conception de notre étude. C'est par respect de cette méthode et d'une logique naturelle, que nous avons découpé notre paragraphe en deux parties importantes:

- l'une examine les besoins à satisfaire, les objectifs à atteindre, les contraintes à respecter. Elle présente les travaux de la concurrence et l'état de la littérature à ce sujet. Nous proposons à la fin de cette étape, et compte-tenu du travail effectué, une orientation possible de l'étude, d'où sont issues les propositions de solutions.
- l'autre, au travers de l'étude préalable, présente puis évalue les différentes solutions et met en évidence les avantages de celles choisies. Elle définit les concepts employés ainsi que le planning. Elle livre une petite partie de la phase "étude détaillée" afin d'illustrer nos choix finaux (langage de programmation, planning révisé de réalisation).

Mais il nous a semblé opportun de compléter ces deux paragraphes par les aspects relationnels et les difficultés que nous avons rencontrées durant cette phase de conception ainsi que les enseignements tirés.

III.3.0 AU DEPART, UN SIMPLE PROBLEME DE MAINTENANCE

A l'origine, nous avons été confrontés à un problème de maintenance rentrant dans le cadre habituel de "suivi de clientèle du SGBD CLIO". Empruntant le cycle de gestion des problèmes CLIO, il est devenu le projet "CLIOPS".

En effet, un problème de maintenance rentrant dans ce cycle de gestion de problèmes (cf fig III.3.0; extrait du travail de formalisation du suivi de clientèle du SGBD CLIO de [JAPS90]) peut déboucher, après sa prise en charge, sur une étude complémentaire. Compte-tenu de la pertinence, de l'ampleur et de la complexité du problème à résoudre, cette étude complémentaire peut elle-même aboutir à une proposition d'évolution du logiciel CLIO.

Ce fut donc l'histoire du projet "CLIOPS"...

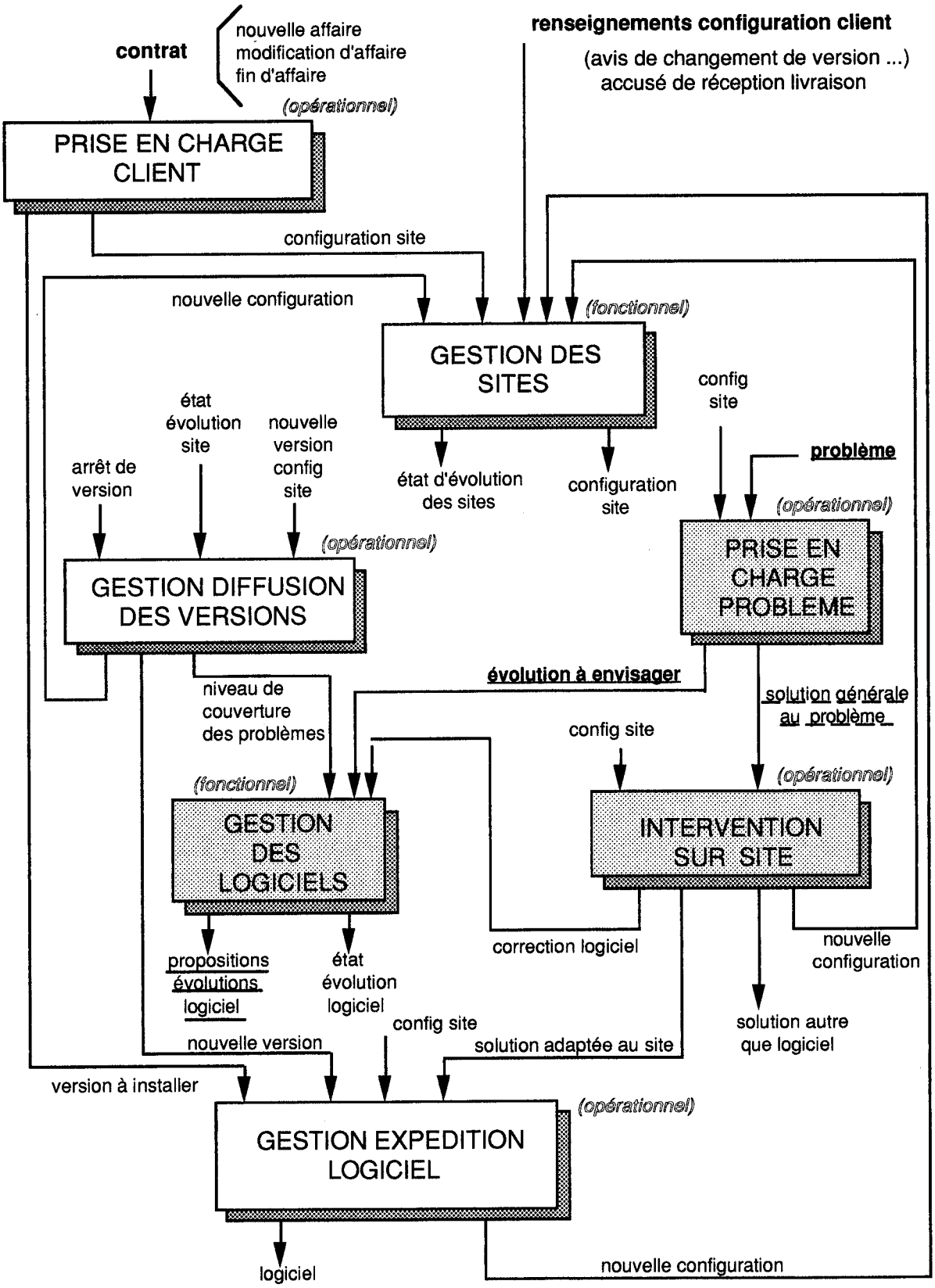


Fig III.3.0 : Suivi de clientèle du SGBD CLIO

III.3.1 BESOINS; OBJECTIFS; CONTRAINTES; CONCURRENCE ET LITTERATURE; ORIENTATION

Compte tenu de la méthodologie choisie, c'est au travers de la prise en charge et du rapport d'observation de l'étude préalable, qu'apparaissent l'analyse des besoins à satisfaire, des objectifs à atteindre et des contraintes à respecter ainsi que la proposition d'une orientation.

III.3.1.1 Les besoins

Le premier point, préconisé par Merise et qui nous a semblé important dans cette phase de conception, fut d'effectuer une étude des besoins en deux phases:

- la première phase de "recueil": il s'agit d'inventorier le plus exhaustivement possible les besoins et désirs (observation de l'existant)
- la deuxième phase "d'évaluation": il s'agit de sélectionner les besoins prioritaires en essayant de les regrouper afin de mieux déterminer nos objectifs.

Complétant cette étude des besoins initiaux, nous avons recherché les besoins "mineurs" succeptibles d'être satisfaits par nos objectifs (prioritaires).

III.3.1.1.1 Expression des besoins

Le bilan de cette phase "recueil" des problèmes et désirs peut se résumer par la mise en évidence de quatre besoins (problèmes) de diverses importances:

1) Problèmes de réseaux

Il s'agit du problème initial que cherche à résoudre notre étude et qui correspond à la première difficulté de maintenance soulevée: nous avons constaté chez plusieurs clients, des anomalies de CLIO (JQ anormalement libéré) suite à des incidents de réseau, erreur significative d'une restauration non faite après un incident. Ce problème nouveau semble se manifester de plus en plus souvent (toute proportion gardée) et peut correspondre à l'accroissement des configurations en réseau (cf II.3).

Nous pouvons le décrire rapidement : l'erreur survient à la suite d'une "coupure de ligne" ou d'une difficulté sur un Decserveur (serveur de terminaux Digital) ou encore d'une chute de réseau ...

Le schéma suivant figure un exemple de réseau avec les éléments qui le composent :

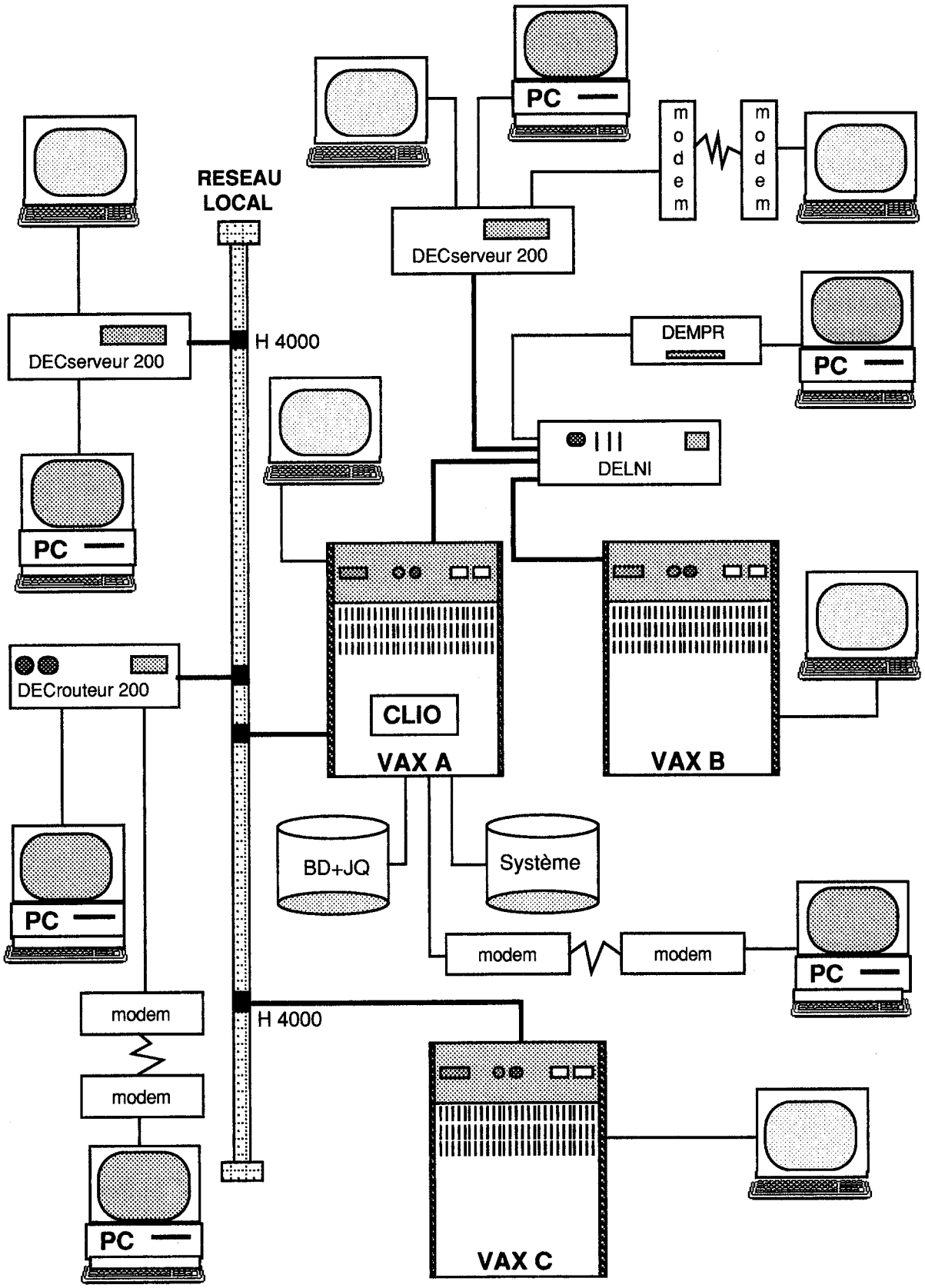


Fig III.3.1.1.1.1 : exemple de réseau DIGITAL avec les éléments qui le composent susceptibles de tomber en panne

En effet, au moment de la détection d'un incident "réseau", le système d'exploitation VMS (Virtual Memory System) semble résoudre le défaut en procédant purement et simplement à une interruption brutale du processus utilisateur. Il ne met donc pas en oeuvre ses mécanismes standards de sécurité qui permettent d'ordinaire, dans les autres types d'incidents (coupure secteur, arrêt disque...), d'activer la sécurité propre à CLIO (restauration des bases de données par utilisation des journaux "avant" ou "rapide" ...).

CLIO, ainsi interrompu, ne peut donc pas garantir l'intégrité et la cohérence de ses informations. En effet, le principe de la sécurité actuelle de CLIO est basé sur la déclaration et l'activation d'un "exit-handler" (cf III.1.2) permettant de débrancher CLIO avant la fin de toute session sur une procédure spéciale (pré-définie lors de la déclaration de l'exit-handler au début de la session CLIO).

Cette procédure spéciale a pour mission d'activer la restauration du journal de sécurité (journal rapide: JQ) de l'utilisateur ayant eu un problème, cela afin d'assurer la cohérence de la base. Ce mécanisme de restauration est un mécanisme classique, identique à ceux présentés dans notre étude théorique générale (cf II.4.2 et III.1.2). Dans le cas de ces incidents réseaux, VMS n'active pas l'"exit-handler" et, par là-même, ne débranche pas CLIO dans sa procédure de sécurité standard.

2) Cluster (grappe d'ordinateurs):

Un "Cluster" est un ensemble de machines reliées par un "bus" rapide ou un réseau local. Il permet de faire partager à ces machines un ensemble de ressources communes. Il donne la vision à l'utilisateur d'une seule grosse machine composée de plusieurs "CPU". L'utilisateur peut demander au système de travailler sur la machine la moins chargée. Le schéma ci-après illustre cet environnement "cluster".

On a pu constater, chez certains clients utilisant des clusters, des difficultés de reconnexion à CLIO, à la suite de l'arrêt d'ordinateur membre d'un cluster. Cet arrêt provoque la réorganisation du cluster et entraîne parfois une mauvaise déconnexion des utilisateurs. Il faut garder à l'esprit que le cluster correspond à un aboutissement en matière d'une politique "réseau" de Digital (cf II.3).

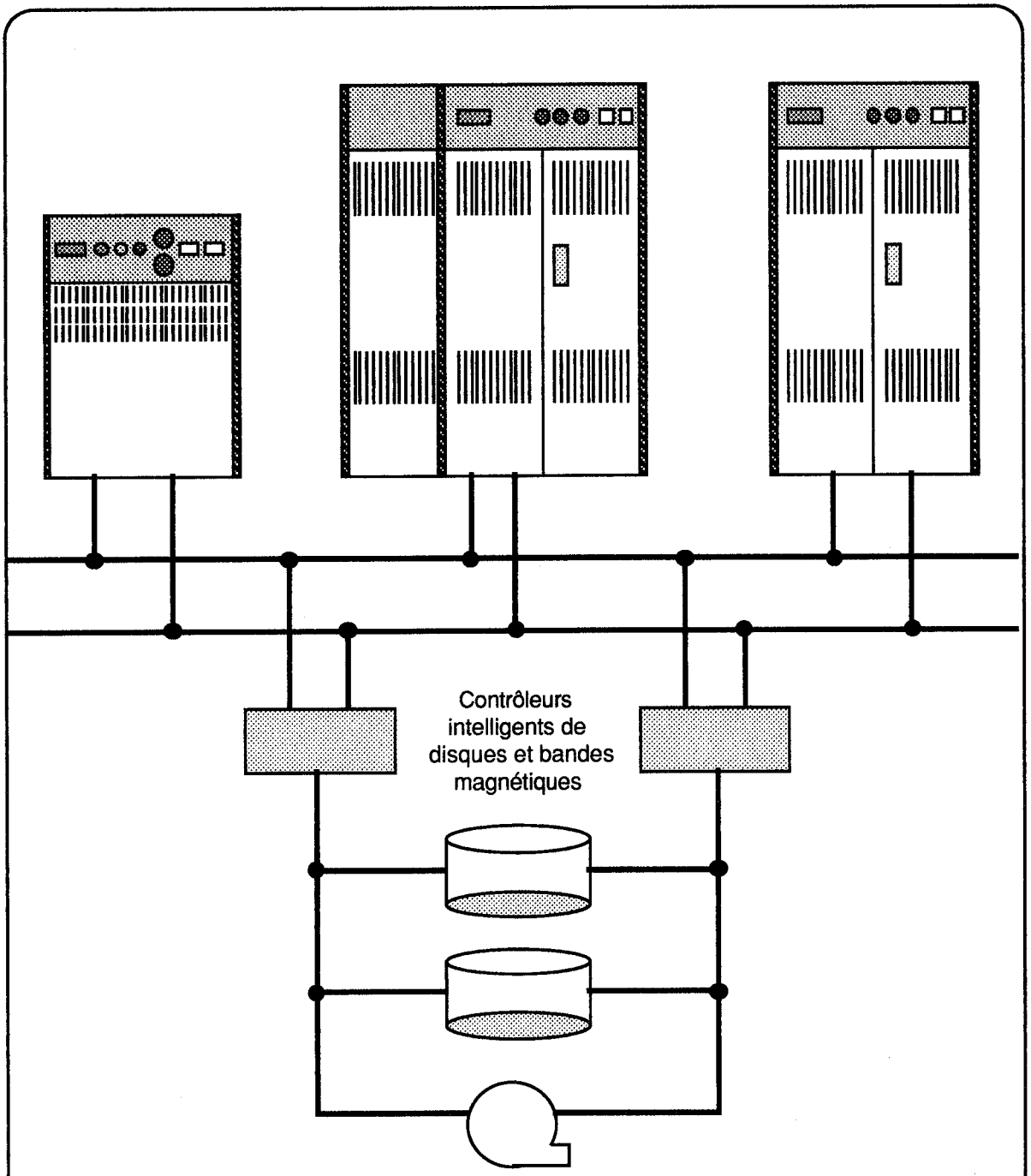


Fig III.3.1.1.1.2 : VAX cluster

3) Amélioration de l'outil d'arrêt des utilisateurs CLIO : "CLIOEXIT"

Notre outil d'arrêt des utilisateurs CLIO "CLIOEXIT" semble dysfonctionner dans les cas très limites. De plus, les utilisateurs souhaiteraient posséder plus d'informations de la part de l'utilitaire et bénéficier de plus de convivialité.

On peut citer un exemple d'anomalie rencontrée: "CLIOEXIT" ne semble pas effectif quand l'utilisateur a sur son clavier la touche "écran figé" opérante. En effet, "CLIOEXIT" envoie l'ordre à VMS d'arrêter CLIO, en effectuant un "exit", qui a pour effet de dérouter CLIO (avant son arrêt) dans une procédure de sécurité prédéfinie lors de la déclaration d'un "exit-handler" au début de la session CLIO. VMS dans ce cas précis ne semble pas pouvoir activer l'"exit-handler" et par là-même la sécurité CLIO.

4) Désir de performances :

Certains de nos clients désirent augmenter les performances de leur application. Ainsi demandent-ils plus d'informations pour mieux évaluer les performances de chaque traitement (qui fait quoi et comment ?)

III.3.1.1.2 Diagnostic de la situation: Evaluation des besoins

Après la phase de détermination exhaustive des besoins, nous avons essayé de procéder à un diagnostic de la situation et à un regroupement de ces besoins. L'évaluation du risque du problème initial (de réseau) avait été effectuée dès le départ. Elle correspondait à l'élément déterminant de l'étude mais elle fut peaufinée dans cette phase de diagnostic.

On a pu déterminer le contexte exact de l'anomalie qui provoque une éventuelle incohérence de la base CLIO, par suite de la non activation de la sécurité. Ainsi, nous avons pu évaluer le risque et les conséquences. Le risque encouru correspond à un incident rare entraînant exceptionnellement de graves conséquences (base cassée). Celles-ci dépendent du contexte de l'anomalie : il faut, pour qu'il y ait risque de base cassée, que la restauration du Journal Q, non faite, ait été réellement nécessaire. Or, pour que cette restauration soit nécessaire, il faut qu'il y ait eu mise à jour disque de la base entre les deux points de cohérence (faute de "page" de la mémoire de l'utilisateur). Mais si l'utilisateur est en attente sur un point de cohérence, alors son JQ est vide, et il n'est donc pas vital de le restaurer. On peut difficilement diagnostiquer les conséquences: on sait que la sécurité n'a pas été activée (JQ anormalement libéré) mais on ne sait pas si elle était réellement utile. Seul l'examen approfondi du journal Q, qui nécessite un spécialiste, ou les statistiques CLIO sur la base (manipulation parfois longue), permettent de le savoir. Il s'agit donc d'un problème gênant car il est difficile à identifier et génère un dilemne. Il convenait, par conséquent, de poursuivre cette étude.

De plus, il nous a paru possible de regrouper des incidents ayant semble-t-il les mêmes origines : les incidents réseaux et les problèmes du "CLIOEXIT", dûs tous deux à la non activation de la sécurité standard de CLIO (basée sur un "exit-handler").

Ainsi, nous pouvons identifier les besoins :

* un besoin prioritaire qui demande que CLIO garantisse la sûreté de fonctionnement (par là-même l'intégrité des données) lorsque les mécanismes standards (basés sur la déclaration d'un exit-handler) sont pris à défaut.
* un besoin secondaire qui correspond à l'obtention de davantage d'informations sur les utilisateurs et de convivialité de l'outil d'arrêt, ainsi qu'un fonctionnement de CLIO dans un environnement "cluster" total.

III.3.1.1.3 Formalisation du fonctionnement actuel de la sécurité CLIO et mise en évidence de ses limites

Le schéma suivant décrit l'enchaînement des opérations du mécanisme actuel de la sécurité CLIO. Il s'agit du modèle conceptuel des traitements existants (MCT existant) issu de notre étude. Nous voyons que le principe de la sécurité actuelle est basé sur la déclaration et l'activation d'un "exit-handler" permettant de se débrancher sur une procédure spéciale (pré-définie lors de la déclaration de l'exit-handler). Cette procédure spéciale a pour mission d'activer la restauration du journal de sécurité (journal rapide: JQ) de l'utilisateur confronté à un problème, cela afin d'assurer la cohérence de la base (mécanisme classique de restauration: cf II.4.2 et III.1.2).

Remarque: il s'agit là de la description d'un incident d'un utilisateur CLIO entraînant la sécurité dite "à chaud" (restauration automatique). Nous n'abordons pas la restauration "à froid" (différée) provoquée par l'arrêt de la machine et stoppant de façon simultanée tous les utilisateurs, car ce mécanisme "RBALL" s'avère, à ce jour, tout à fait satisfaisant (cf II.4 et III.1.2).

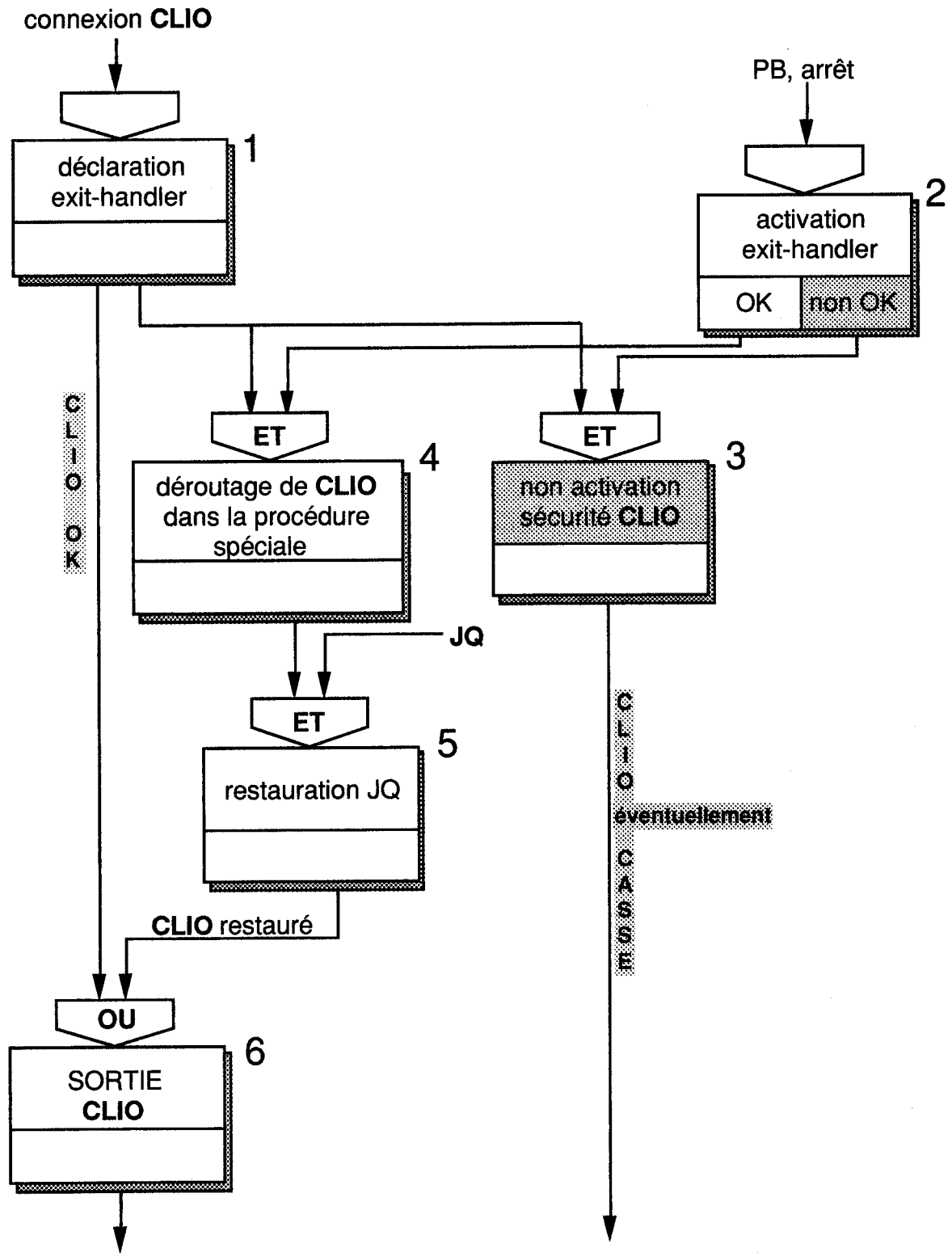


Fig III.3.1.1.3 : (MCT actuel) visualisation du fonctionnement actuel de la sécurité CLIO. Mise en évidence des limites de ce mécanisme

III.3.1.2 Objectifs

Notre méthode fut de déterminer un objectif permettant de répondre au besoin prioritaire, tout en gardant à l'esprit que cet objectif devait rester approprié aux autres besoins secondaires.

Le besoin prioritaire a été examiné dans le schéma III.3.1.1: la phase critique mise en évidence dans ce schéma est l'opération 3, c'est-à-dire un incident n'activant pas "l'exit-handler" et par là-même la sécurité standard de CLIO. Il faut donc effectuer une substitution: ajout d'un mécanisme supplémentaire et remplacement de l'opération 3 inopérante par l'opération 3 BIS "active". Il s'agit du modèle conceptuel des traitements proposés (MCT proposé). Il en résulte que CLIO est dérouté vers une opération "qui le restaure". Elle permet de récupérer le flux correct de CLIO :

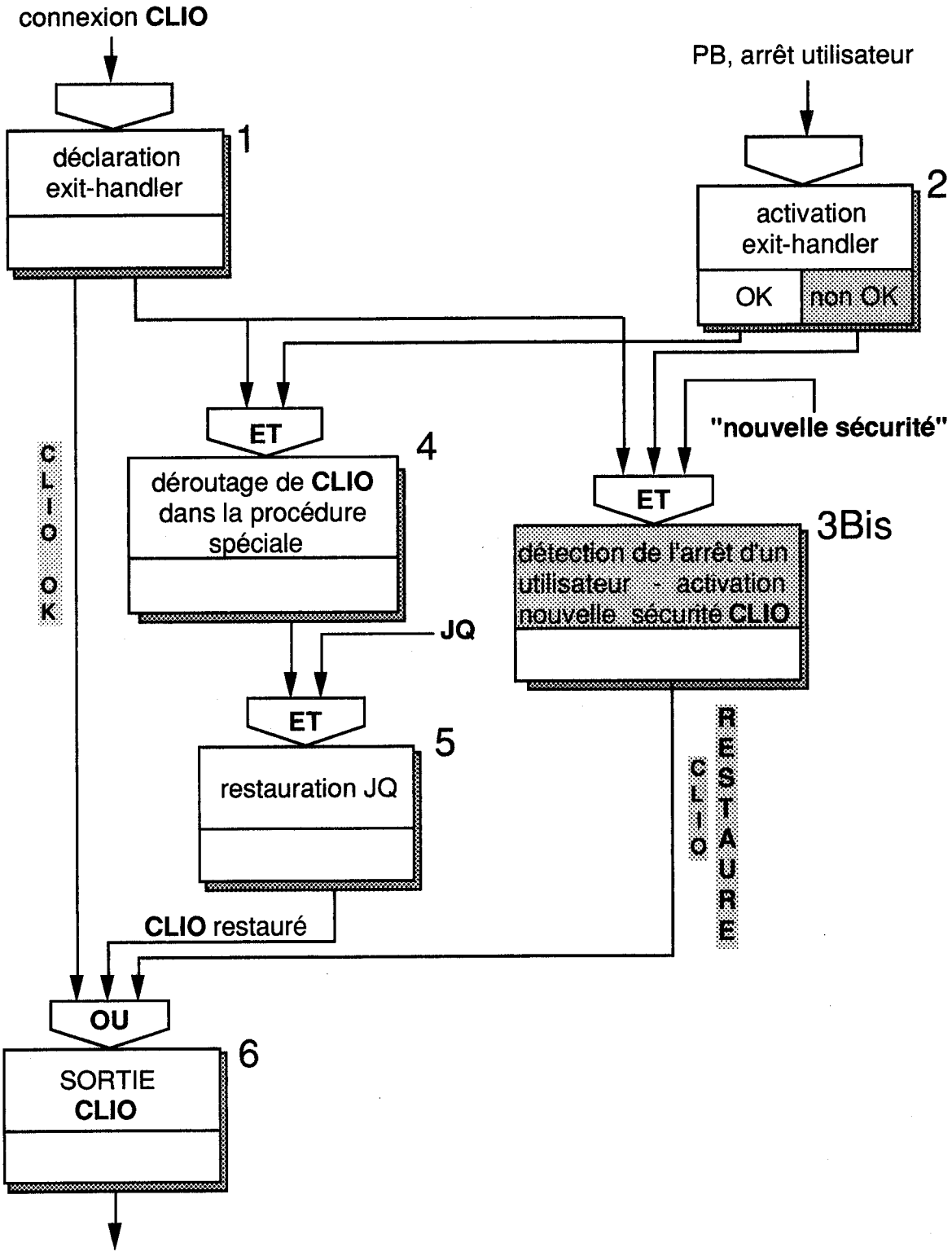


Fig III.3.1.2 : (MCT) visualisation du fonctionnement de la sécurité CLIO avec l'adjonction d'une nouvelle sécurité

Il s'agit maintenant de savoir si l'objectif prioritaire peut satisfaire les autres besoins. La nécessité d'utilisation en cluster ne semble pas incompatible avec l'objectif d'assurer un fonctionnement sans faille en environnement réseau : il suffira de modifier ce mécanisme mono-machine afin qu'il sache communiquer avec les autres membres de la grappe d'ordinateurs.

Le désir de plus d'informations concernant les utilisateurs ne semble pas non plus contradictoire avec le premier objectif, mais peut constituer une fonction supplémentaire.

En conclusion, notre objectif prioritiaire consistera donc à réaliser un mécanisme de base permettant de détecter toute "non activation de la sécurité standard" afin d'assurer un parfait fonctionnement de CLIO en environnement réseau. Ce mécanisme de base devra être réutilisable pour un nouveau "clioexit", et à plus long terme permettre un fonctionnement de CLIO en cluster ainsi qu'une délivrance d'informations sur CLIO et ses utilisateurs.

III.3.1.3 Contraintes

Après avoir déterminé les objectifs de notre étude, il convient d'indiquer les contraintes que nous nous sommes fixées.

Nous avons toujours eu à l'esprit que CLIO/VAX est un produit déjà largement diffusé, aux performances reconnues et aux clients attentifs qui n'en n'accepteraient pas la dégradation. Aussi, les "contraintes clients" sont-elles les premières à être envisagées impérativement.

"contraintes clients":

- CLIO doit rester performant, c'est-à-dire l'ajoût du nouveau mécanisme de sécurité ne doit pas pénaliser l'utilisateur
- la solution ne doit pas remettre en question le produit; il s'agit d'un apport important mais sans refonte totale
- la nouvelle sécurité doit être simple de mise en oeuvre.

Le projet, correspondant à une étude interne Syséca, implique aussi d'autres contraintes:

"contraintes internes"

- la conception et la réalisation de la solution se doivent de ne pas consommer plus d'un certain nombre d'hommes mois fixé par la direction
- le projet, et surtout la phase de conception, doivent pouvoir être supervisés par le responsable de l'étude interne (remise de documents indiquant l'évolution de l'analyse)
- la solution doit respecter le planning sans débordement
- la solution doit être ouverte, évolutive, adaptable et réutilisable pour d'autres équipes
- la solution doit être facilement et rapidement intégrable au CLIO existant.

Des contraintes sont aussi générées par nos objectifs:

"contraintes de l'objectif"

il faut que le mécanisme mis en oeuvre par l'objectif prioritaire de "sûreté de fonctionnement", corresponde à un mécanisme de base réutilisable pour nos objectifs à plus long terme : le "cluster" et la "délivrance d'informations".

III.3.1.4 Etude de la concurrence (rdb,oracle...) et le point sur la littérature:

Cette recherche m'a paru indispensable afin d'élaborer notre solution; il semblait primordial de se poser certaines questions:

- les concurrents sont-ils confrontés aux mêmes problèmes? Quelles solutions ont-ils choisi ?
- la "littérature informatique" soulève-t-elle le problème de la sécurité en environnement réseau ?

En ce qui concerne la "littérature informatique", tous les auteurs n'analysent pas ce problème ([ADIB83] [GARD83] [COHEN89] [AKOKA84] [ABDE90]), mais ceux qui l'abordent ([GALACSI89] [PHI86] [GARVAL85] [PASL90]) nous indiquent que la sécurité activée semble basée sur l'utilisation d'un journal comme dans une sécurité classique (cf:II.2).

Evoquons maintenant la concurrence: là encore, il a été difficile de trouver des informations précises. Les auteurs restent flous: ils proposent un mécanisme de sécurité mais sans le dévoiler davantage. Il apparaît tout de même que "Oracle" semble gérer ces problèmes de défaillance de réseau:

[PHI86] nous explique: <<s'il s'agit d'une coupure de ligne, l'état de la base est restitué. On peut, par sécurité, lancer une procédure de Rollback>>.

[MINI89] précise la question: <<le processus "Cleanup" (CLN) détecte toute fin anormale de processus et gère la reprise après incident>>.

De même, [RDBa],[RDBb] nous informent que "RDB", le SGBD de Digital, possède aussi un mécanisme de surveillance "RDMS_MONITOR" utilisé dans les environnements réseaux et surtout Cluster assurant "l'automatic database recovery with before journaling" (RUJ: Recovery Unit Journal) et rétablissant automatiquement l'intégrité de la base de données de façon transparente ("failover transparent"). Ils précisent qu'il s'agit d'un process détaché, ("Monitor Process"), qui a pour mission de contrôler également les accès à la base.

Il existe avec le CLIO distribué la notion de Surveilleur. Ce Surveilleur des machines serveuses assure l'intégrité de CLIO dans le cas où un processus serveur viendrait à disparaître. Le surveilleur restaure le "JQ" (cf:III.1.2) du processus disparu afin d'assurer l'intégrité de la base CLIO.

L' étude de la littérature informatique et de la concurrence nous amène donc à penser que, pour détecter l'arrêt anormal des utilisateurs, l'emploi d'un mécanisme de surveillance semble s'imposer. Ce concept de surveillance correspond à un processus de sécurité capable de rejouer le journal de sécurité (journal avant) de l'utilisateur afin d'assurer l'intégrité de la base .

III.3.1.5 Orientation choisie:

Ayant déterminé les besoins, les objectifs, les contraintes et abordé la concurrence, il convient de proposer une orientation sur laquelle devront se baser toutes les solutions possibles.

Il se dégage de notre analyse de la concurrence que l'utilisation d'un processus de surveillance est envisagée, et de plus, nous notons que ce principe est déjà employé par le CLIO distribué. A ce stade, l'orientation à prendre nous paraît donc claire: il s'agit, pour nous aussi, de pourvoir CLIO/VAX d'un processus de surveillance apte à connaître toutes les connexions et déconnexions à CLIO. Ce processus de surveillance détecte les déconnexions intempestives et peut prendre les dispositions adéquates pour assurer la cohérence de la base (restauration du journal de sécurité de l'utilisateur).

De plus, possédant ce système central, il est possible de coupler le processus de surveillance à "CLIOEXIT" (cf:III.3.1.1.3) ainsi qu' à un outil de "monitoring" (évaluation).

Nous créons un mécanisme, pour qu'à long terme, nous puissions utiliser pleinement CLIO en environnement "Cluster", grâce à l'emploi de plusieurs processus de surveillance reliés entre eux (un par machine du "Cluster").

Le principe de "CLIOPS" (CLIO Processus de Surveillance) était né, il restait à proposer des solutions techniques.

L'orientation choisie est illustrée par le schéma suivant :

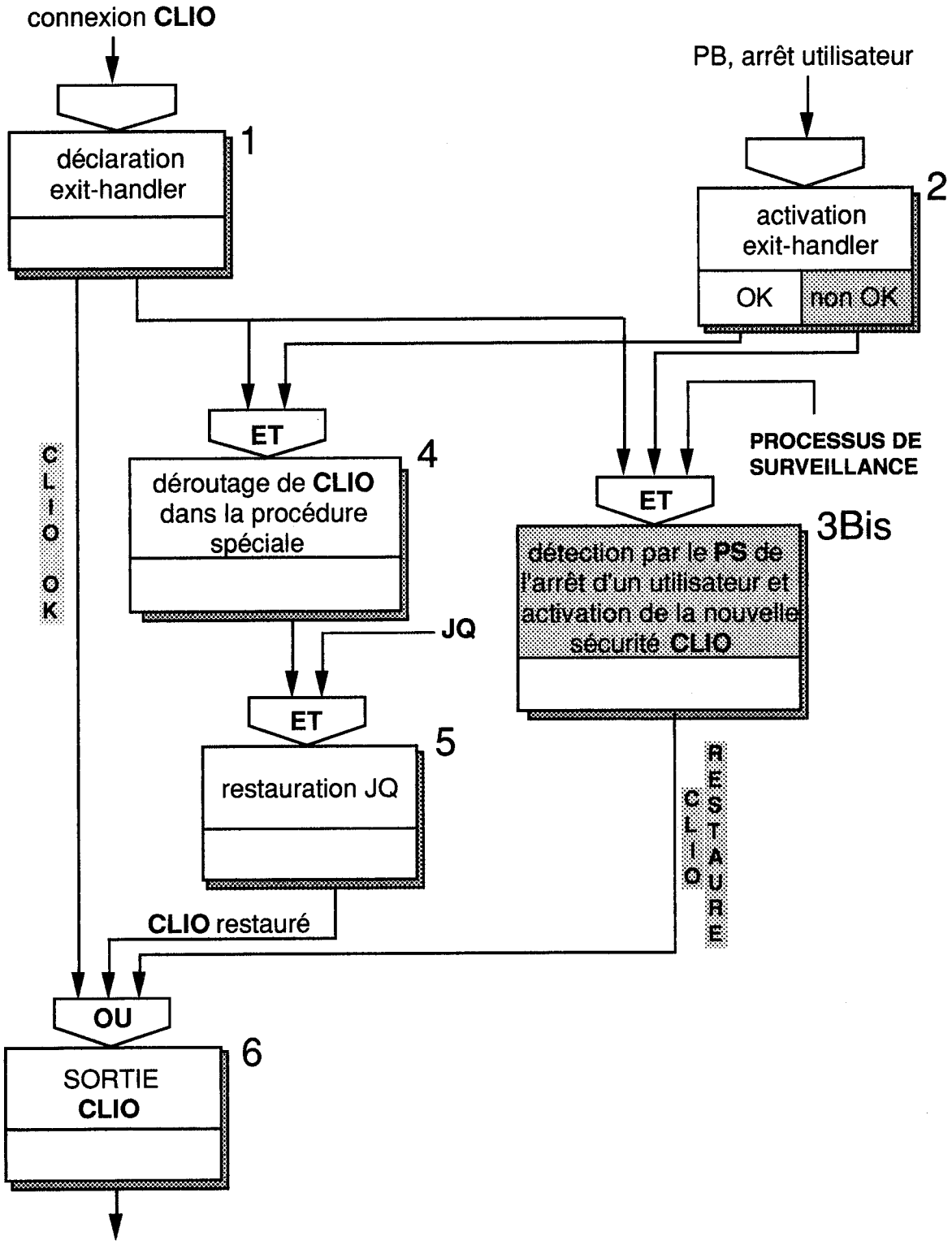


Fig III.3.1.5 : (MCT) visualisation du fonctionnement de la sécurité CLIO avec l'adjonction du Processus de Surveillance

III.3.2 SOLUTIONS POSSIBLES ET CHOIX; CONCEPTION DE LA SOLUTION

Ce paragraphe est constitué de deux parties: la première évalue, au travers de l'étude préalable, les différentes solutions possibles, mettant en évidence les avantages de la solution choisie et expose le planning. Dans la seconde partie, nous présentons un aperçu de la phase de l'étude détaillée afin d'illustrer nos choix finaux (langage de programmation, planning révisé de réalisation).

III.3.2.1 Etude Préalable

III.3.2.1.1 Recherche et élaboration des solutions possibles

L'expérience du CLIO/DISTRIBUE :

Possédant au sein de CLIO des compétences en matière de Processus de Surveillance avec CLIO/DISTRIBUE, le bon sens voulait que j'aille m'entretenir avec le responsable de cette équipe afin d'analyser leur Surveilleur et de vérifier s'il n'était pas possible de le réutiliser ou du moins de s'en inspirer.

En résumé, le Surveilleur (sur machine UNIX) des Machines Serveuses effectue un tour de garde toutes les dix secondes: si un utilisateur MS (Machine Serveuse) n'existe plus ("arrêté anormalement"), le surveilleur cherche à "rollbacker" (restauration à chaud) son JQ. Le surveilleur connaît le numéro du JQ grâce à sa table d'informations. Une fois le journal restauré (c'est-à-dire lorsque l'utilisateur se retrouve à nouveau dans un état cohérent), le surveilleur libère les verrous (sur la page de données) détenus par l'utilisateur "tombé". Mais ceci n'est valable qu'avec CLIO UNIX qui possède son propre GAC (Gestionnaire d'Accès Concurrents) ou gestionnaire de verrous.

Le LOCKMANAGER de VMS (GAC des VAX de Digital et donc de CLIO VAX) libère les verrous dès que l'utilisateur est "tombé" : il serait donc impossible pour le Surveilleur de "Rollbacker" le JQ puisqu'il ne sait pas si les verrous anormalement libérés (lors de la chute de l'utilisateur par le "Lockmanager") n'ont pas été réattribués. Le "rollback" du JQ de l'utilisateur "tombé" pourrait remodifier des données (pages) déjà mises à jour par le deuxième utilisateur qui s'est alloué les verrous anormalement libérés lors de la chute du premier.

Principes communs à toutes les solutions possibles:

Le surveilleur du CLIO/DISTRIBUE sous UNIX ne semble donc pas récupérable tel quel. Mais certaines idées m'ont semblé exploitables et m'ont aidé à déterminer les principes de bases du Processus de Surveillance de CLIO/VAX.

Au delà des objectifs fixés, les solutions possibles doivent donc respecter les données suivantes:

- le processus de surveillance est un processus indépendant qui doit pouvoir détecter l'arrêt d'un process utilisateur CLIO et restaurer le contexte correct de cet utilisateur (journal de sécurité) c'est-à-dire assurer la cohérence de la base
- l'utilisateur CLIO doit pouvoir détecter l'arrêt du Processus de Surveillance (PS)

Il en découle des principes :

1) utilisation d'une table d'informations:

Une table d'informations (en mémoire) mise à jour sur chaque connexion ou déconnexion des utilisateurs par le Processus de Surveillance, ou par les utilisateurs eux-mêmes, contient des informations sur les différents utilisateurs connectés à une base CLIO (processus identificateur (PID), numéro d'utilisateur CLIO (n°UTI), nombre total d'utilisateurs (nbUTI), nom de la base sur laquelle l'utilisateur est connecté etc...)

2) communication entre PS et utilisateur CLIO (par boîte aux lettres)

3) restauration par le PS du JQ de l'utilisateur arrêté anormalement

4) "rollback" (retour arrière) effectué par les autres utilisateurs

Compte-tenu de ces obligations, il en résulte que les solutions possibles ne pourront se différencier que sur ces points:
- la façon (la manière et le moment) de détecter l'arrêt d'un utilisateur
- le rollback des utilisateurs (suite à leur initiative ou à un ordre du PS).

Les différentes solutions possibles:

Nous avons pu présenter trois solutions. Elles consistent toutes à pallier les failles créées par une hypothétique utilisation du Surveilleur de CLIO/DISTRIBUE. En effet, il faut que les utilisateurs CLIO ne puissent pas utiliser les pages (ressources,données) attribuées anormalement lors de la libération des verrous associés .

Les deux premières solutions partent du principe que c'est l'utilisateur qui détecte l'arrêt anormal d'un autre utilisateur, et qui s'interdit de s'attribuer ces ressources en invalidant ses propres demandes. De plus, c'est donc lui qui prend l'initiative du "rollback". Les deux premières solutions se distinguent par le choix du moment de l'action.

Solution "ressource":

Elle est basée sur la vérification au moment d'une demande de ressource (page à laquelle est associée un verrou) de la présence de tous les utilisateurs CLIO. Lors de la demande de ressource (verrou associé) par un utilisateur, celui-ci parcourt la table des utilisateurs (TABLE PS) et, pour chacun d'eux, vérifie s'il ne s'est pas arrêté anormalement (sans prévenir c'est-à-dire sans mise à jour de la table). S'il constate qu'un utilisateur référencé dans la table n'existe plus, il annule sa demande de verrou et prévient le PS (Processus de Surveillance).

Solution "CKPT" (check point):

Elle reprend le même principe mais la vérification s'effectue au moment de la prise d'un point de contrôle (check-point cf III.1.3.2). L'utilisateur incertain quant à ses demandes de ressources, invalide son travail (prise de ressource) par un "rollback".

Solution "verrou":

Ce ne sont plus les utilisateurs qui détectent l'arrêt anormal de l'un d'entre eux mais cette fonction revient au PS. Cette solution peut se résumer ainsi: à sa connexion, chaque utilisateur prend un verrou (en mono) l'identifiant "CLIO_<PID>" (PID correspond à son process identificateur). Le PS averti de cette connexion, cherche à obtenir le même verrou (demande sans attente mais avec interruption en cas d'obtention (ast: cf III.3.2.1.3.a.4)). Si un utilisateur s'arrête anormalement (sans avoir demandé de déconnexion au PS), il libère son verrou CLIO_<PID>. Ce verrou est attribué immédiatement au PS qui en était demandeur. Ainsi celui-ci sait si tel ou tel utisateur est "tombé". Il envoie alors à tous les utilisateurs CLIO l'ordre de se "rollbacker" (de même type qu'un "deadlock") via une boîte aux lettres de communication. Les processus utilisateurs CLIO recevant ce message se "rollbackent" (retour au dernier point de cohérence (CKPT)) et par là-même invalident les pages qu'ils n'auraient pas dû s'attribuer.

III.3.2.1.2 Evaluation des solutions possibles , et choix:

Après avoir proposé différentes solutions, il convient d'effectuer un choix; le tableau suivant illustre notre évaluation:

	mise en oeuvre du mécanisme compte tenu de la rareté de l'incident	mode de détection	type de mécanisme	évaluation coût du mécanisme	sûreté du mécanisme
solution **"ressource"**	très souvent	test (existence utilisateur)	appel fonction système	<taille table> x appel système x nb demande verrou	sûre si pas d'incident pendant le parcours table
solution **"CKPT"**	souvent	test (existence utilisateur)	appel fonction système	<taille table> x appel système x nb demande CKPT	sûre si pas d'incident pendant le parcours table
solution **"verrou"**	quand il faut : rare	attribution de verrou (clio <pid>)	ast (asynchorus system trap)	nombre d'incidents	sûre si PS prévient utilisateur avant CKPT sur page libérée anormalement

Fig III.3.2.1.2 : évaluation des solutions possibles

Toutes ces solutions techniques de réalisation respectent les objectifs et contraintes fixés antérieurement. De plus, ces solutions ne sont pas fondamentalement différentes. Seuls les critères de performance et de refonte de CLIO détermineront notre choix. Il semble qu'aucune de ces solutions, par leur spécificité, n'apporte une remise en question totale de CLIO. Il en résulte donc que seul le facteur de performance est décisif. Notre étude indique de façon évidente, que la troisième solution (verrou) est la moins coûteuse.

III.3.2.1.3 La solution retenue: solution "verrou":

Nous pouvons, sans trop détailler car ce n'est pas le but principal de ce mémoire, présenter la solution retenue. Nous rappelons dans un premier temps les notions utilisées tout en les précisant. Nous illustrerons ensuite les mécanismes mis en oeuvre par quelques exemples de modèles conceptuels extraits de notre étude préalable.

a) Rappel et précision des notions utilisées par le PS

-1) Utilisation d'un verrou CLIOPS_<machine> qui sert aux utilisateurs CLIO à détecter l'arrêt du PS :
le principe de cette méthode (détection de l'arrêt d'un process grâce à un verrou) peut se résumer ainsi: à sa connexion, le PS prend un verrou (mono) CLIOPS_<machine> (machine sur laquelle travaille le PS ou plus exactement la Base des Bases). Les utilisateurs cherchent à obtenir le verrou cliops_<machine> (ils demandent à être interrompus s'ils peuvent l'obtenir (AST: cf point 4 suivant). Si le PS s'arrête anormalement, il libère le verrou cliops_<machine>, et les utilisateurs l'obtiennent. Cette obtention permet de savoir que le PS a disparu. Normalement, le PS s'arrête uniquement lorsqu'il n'y a plus d'utilisateurs connectés (actifs).

-2) Utilisation de boîte aux lettres (mailbox) pour la communication entre le PS et les utilisateurs CLIO:
tout utilisateur doit signaler sa connexion ou sa déconnexion au Processus de Surveillance par l'envoi d'un message sur la mailbox de communication. Ce message permet au PS de connaître les utilisateurs connectés et de mettre à jour sa table d'informations (tableps). Le PS peut aussi envoyer des ordres aux utilisateurs via leur mailbox de communication (ex: ordre de se rollbacker suite à l'arrêt anormal d'un utilisateur). De plus, le PS peut recevoir des messages des utilisateurs (ex: compte rendus).

-3) Utilisation d'un verrou CLIOS_<PID> servant à détecter un utilisateur "tombé" (arrêt anormal):
à sa connexion, l'utilisateur prend un verrou (mono) CLIOS_<PID> (PID correspondant à son Processus Idenficateur). Le PS cherche à obtenir le verrou CLIOS_<PID> (il demande à être interrompu s'il peut l'obtenir (AST : cf point 4 suivant)). Si l'utilisateur s'arrête, il libère le verrou et le PS l'obtient, ce qui lui permet de détecter l'arrêt anormal de cet utilisateur. Normalement, l'utilisateur prévient le PS de sa déconnexion et de son projet de libération de son verrou.

-4) Utilisation d'AST (Asynchronus System Trap):
elle permet qu'un processus (utilisateur ou PS) soit interrompu (ex: lorsqu'on alloue à l'utilisateur le verrou demandé ou lorsqu'arrive un message sur sa boîte aux lettres), et dérouté sur l'exécution d'une procédure prédéfinie.

Le schéma suivant illustre les notions précédemment énoncées: le mode de communication entre le PS et les différents utilisateurs (utilisation de boîte aux lettres). Il met en évidence la table d'informations du PS ainsi que l'utilisation de verrous pour la détection de l'arrêt d'un utilisateur ou du Processus de Surveillance:

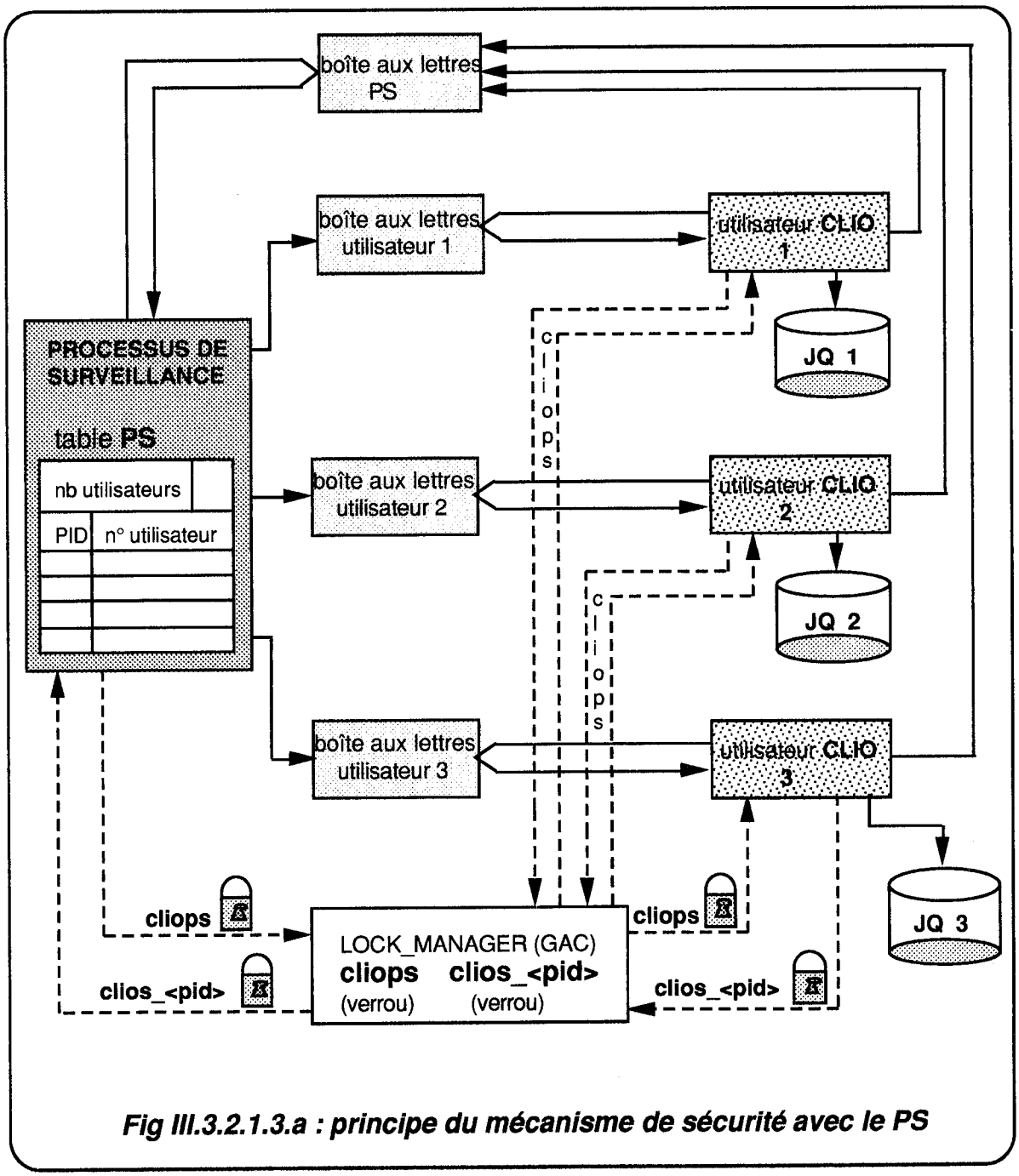


Fig III.3.2.1.3.a : principe du mécanisme de sécurité avec le PS

b) Les mécanismes de CLIOPS

Après avoir rappelé les différents éléments intervenant dans CLIOPS (verrou, boîte aux lettres, ast), il convient de définir le mécanisme. Sa mise en oeuvre intervient à trois "moments":

1) connexion (opération 1) et déconnexion (opération 6) de l'utilisateur

2) substitution de l'opération 3 et 3BIS (détection de l'arrêt d'un utilisateur)

3) connexion, déconnexion et arrêt du PS

Par souci de clarté, la méthodologie suivie nous a invité à décrire les différentes opérations composant les fonctions de notre mécanisme en partant du général, et à ne les détailler en sous-opérations que dans un deuxième temps. On pratique par enrichissements successifs des schémas pour arriver à la solution finale détaillée.

Comme nous l'avons introduit au début de ce paragraphe, nous ne présentons pas tout le travail de formalisation de notre solution. Mais afin d' illustrer tout de même notre méthodologie, nous en donnons quelques exemples de modélisations: nous livrons deux modèles conceptuels de traitement (MCT) généraux figurant d'une part la connexion et la déconnexion d'un utilisateur surveillé par le PS, d'autre part la détection de l'arrêt anormal d'un utilisateur par le PS. Nous terminons par un exemple de MCT approfondissant la détection de l'arrêt anormal d'un utilisateur par le PS, et la mise en oeuvre de la nouvelle sécurité CLIO.

b.1) Connexion , détection de l'arrêt d'un utilisateur:

Décrivons tout d'abord de façon simplifiée (fig.III.3.2.1.3.b.1) le mécanisme de l'opération 3BIS, substitution de l'opération 3 défectueuse de la fig III.3.1.1.3.

Cette opération 3BIS est constituée par deux opérations 3bis1 et 3bis2 qui permettent la libération du verrou clios_<pid_x> de l'utilisateur CLIO X et la détection de cette libération par le PS. La libération de ce verrou est la clef du nouveau mécanisme de sécurité. La nouvelle opération de connexion d'un utilisateur avec le PS (opération 1BIS) permet à cet utilisateur X, ayant comme PID :<pid_x> (process identification), de déclarer et de prendre le verrou clios_<pid_x> nécessaire à la nouvelle fonctionnalité de l'opération 3.

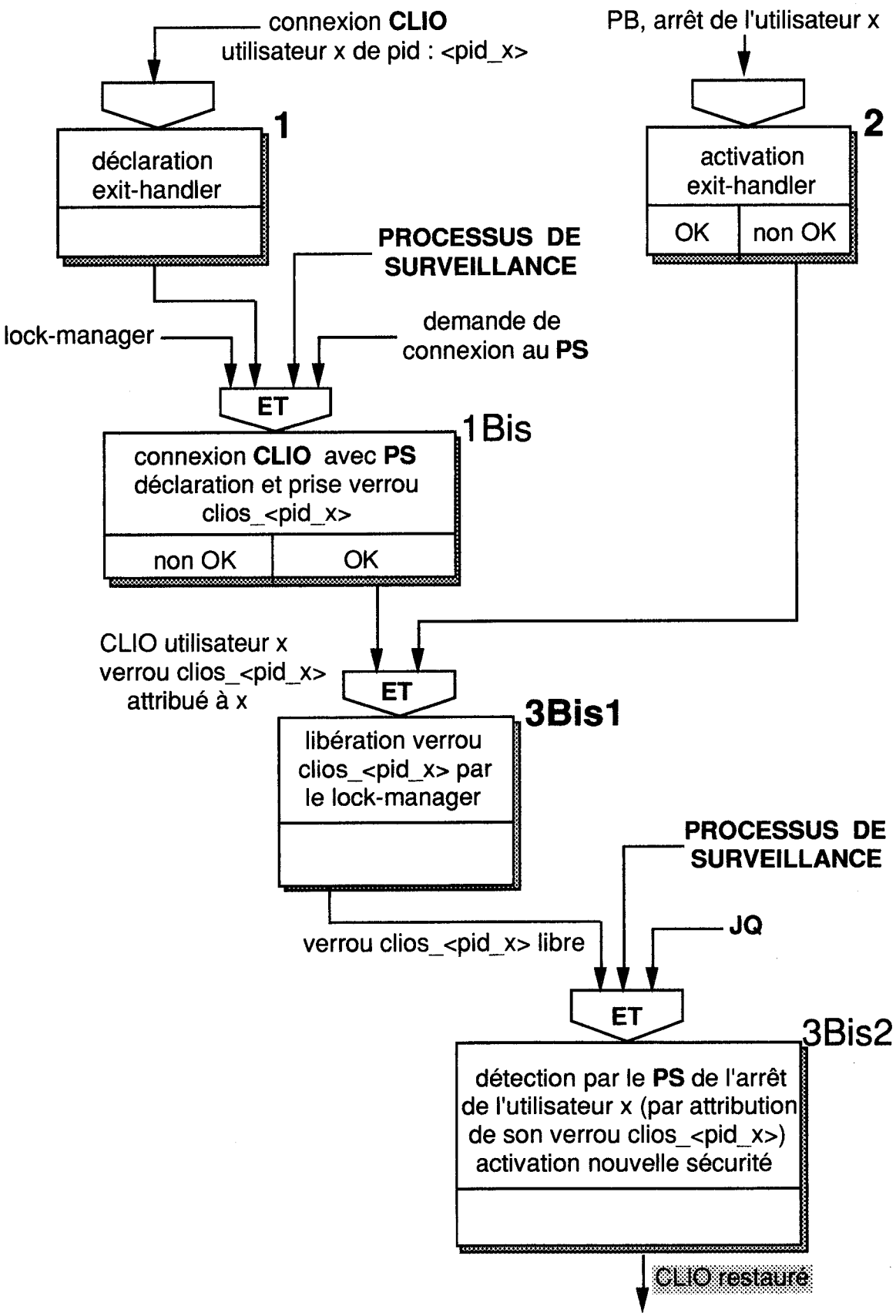


Fig III.3.2.1.3.b.1 : connexion d'un utilisateur, détection de son arrêt et description traitement (opérations 1Bis & 3Bis simplifiées)

b.2) Connexion et déconnexion d'un utilisateur:

Dans le schéma suivant, nous détaillons davantage le mécanisme de connexion (opération 1bis): cela se concrétise par l'ajoût de l'opération 8 (mise à jour de la table du PS). On précise de la même façon le mécanisme de déconnexion illustré par les opérations nouvelles 7 et 8.

Ce schéma met aussi en évidence la manière dont l'ancienne sécurité toujours active (celle activée par l'exit-handler) s'intègre au nouveau mécanisme (l'opération 5 est reliée à la nouvelle opération 7).

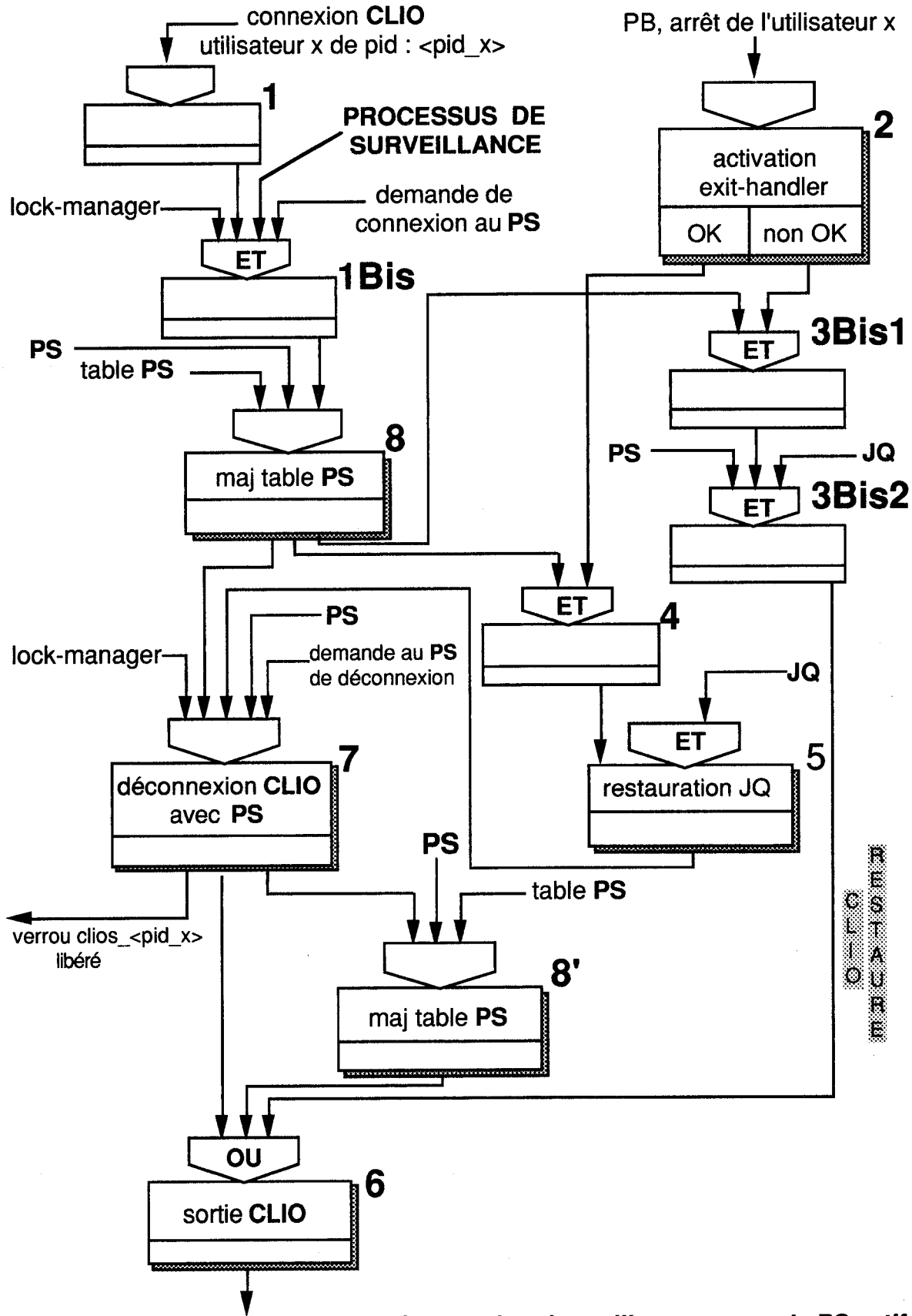


Fig III.3.2.1.3.b.2 : connexion et déconnexion des utilisateurs avec le PS actif

b.3) Exemple d'approfondissement des opérations de la nouvelle sécurité: détection par le PS de l'arrêt d'un utilisateur

L'arrêt d'un utilisateur provoque sa détection par le PS. Ce mécanisme illustré (fig.III.3.2.1.3.b.1) dans le paragraphe précédent, est précisé dans la figure suivante. Cette détection provoque une restauration du JQ de l'utilisateur X tombé, par le PS, et oblige tous les utilisateurs non arrêtés à se "rollbacker". Il nécessite l'ajoût de quatre nouvelles opérations 3bis3, 3bis4, 3bis5 et 3bis6 respectivement "rolllback des autres utilisateurs" (non "tombés"), "restauration du JQ de l'utilisateur "tombé"", "mise à jour de la table PS" et enfin "redémarrage des utilisateurs rollbackés".

La restauration du JQ de l'utilisateur "tombé" ne peut s'effectuer qu'après le rollback (retour arrière au dernier point de cohérence) de tous les autres utilisateurs, c'est-à-dire après la réception par le PS de leur message "rollback_OK" (sur sa boîte aux lettres cliops_<machine>) .

Les utilisateurs "rollbackés" ne "repartiront" (restart) qu'après avoir obtenu du PS le droit de redémarrer (libération du verrou V_CKPT) (3bis6).

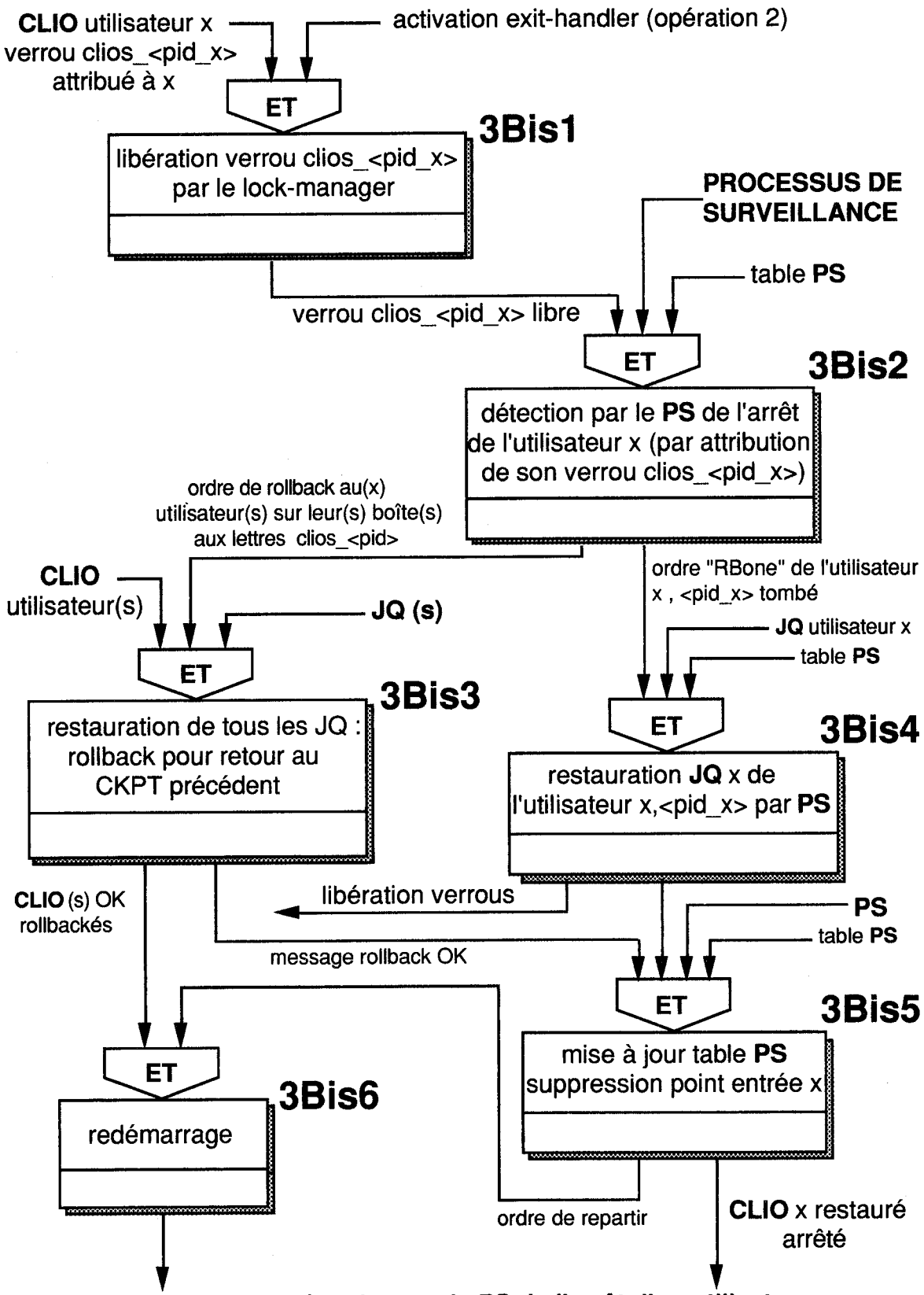


Fig III.3.2.1.3.b.3 : détection par le PS de l'arrêt d'un utilisateur (conception approfondie)

III.3.2.1.4 Echéancier prévisionnel de réalisation (planning)

Parvenu à la fin de notre étude préalable, il convient de revenir sur l'échéancier prévisionnel de réalisation. En effet, la méthode Merise incite à revoir son planning après chaque phase importante afin d'essayer d'anticiper tout débordement.

Rappelons aussi que le respect du planning constitue l'une des préoccupations de l'informaticien cherchant à mener à bien son projet, et que c'est une contrainte que nous nous sommes fixés.

Il sera également réactualisé à la fin de notre analyse.

Le planning originel correspond donc à l'échéancier prévisionnel de réalisation donné lors de notre proposition de sujet de mémoire d'ingénieur CNAM (planning repris pour tenir compte des remarques formulées lors de sa présentation le jour du probatoire).

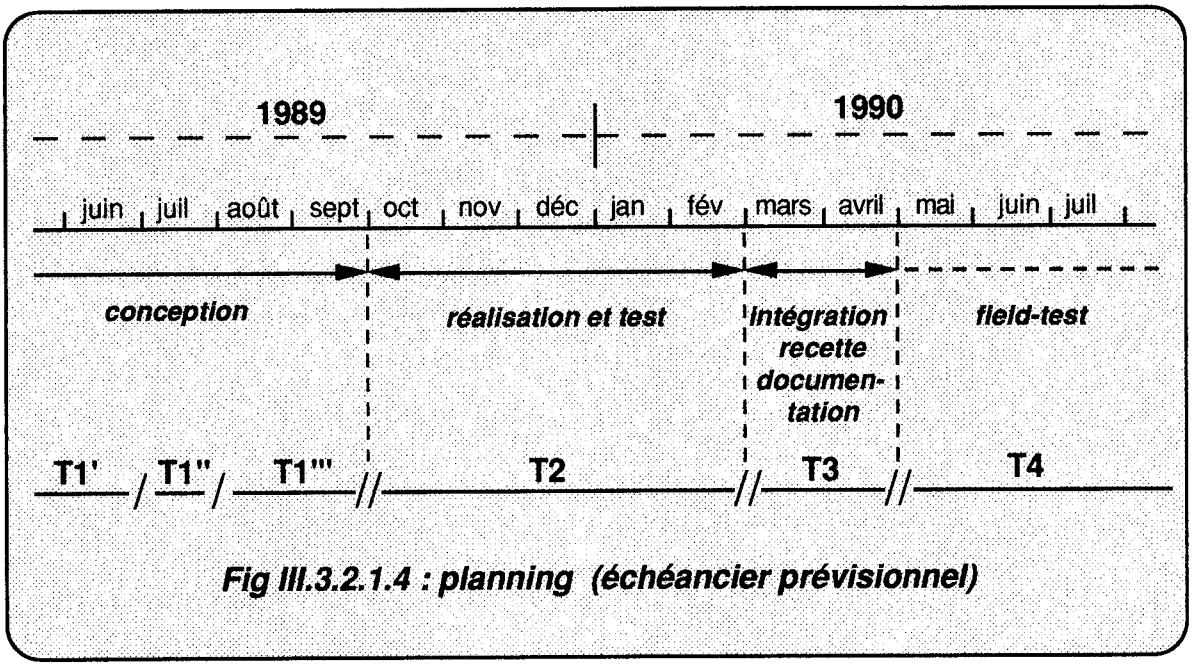


Fig III.3.2.1.4 : planning (échéancier prévisionnel)

III.3.2.2 L'Etude Détaillée

TACHE T1' JUIN 1989

* Etude de la littérature informatique et de la concurrence afin de déterminer l'état de la recherche et les notions en matière de sécurité d'un SGBD et plus particulièrement en environnement réseau informatique.
* Constitution d'un dossier en vue de l'élaboration d'une partie théorique de notre mémoire

TACHE T1" JUILLET 1989

* Pré-spécifications générales
* Rapport d'observations:
  - détermination des besoins, diagnostic de situation
  - détermination des objectifs et contraintes
  - proposition d'une orientation
* Validation de l'orientation
* Proposition de solutions
* choix de la solution

TACHE T1''' AOUT 1989
* Elaboration et validation des spécifications préalables de la solution adoptée
* Ecriture des spécifications techniques détaillées
* Validation du dossier d'analyse
* Elaboration d'un dossier à partir de l'analyse en vue de la rédaction du mémoire

TACHE 2 OCTOBRE 1989
* Préparation et répartition du travail de chacun des membres de l'équipe réalisatrice (objectif, délais, calendrier de réalisation, réunions d'avancement et comptes rendus)
* Réalisation de la maquette
* Tests d'intégration à CLIO
* Suite de l'élaboration du dossier en vue de la rédaction du mémoire, rédaction du mémoire.

TACHE 3 MARS 1990
* Intégration du produit à CLIO
  - Prototype
  - version 1
* Validation interne de l'ensemble du produit
* Elaboration de la documentation complète du logiciel
* Poursuite de la rédaction du mémoire

TACHE 4 MAI 1990
* Essais du produit en pré-exploitation chez un ou plusieurs clients (field-test) de la phase 1 du produit (version 1).
* Détermination de la date d'une livraison définitive
* Mise en forme définitive du mémoire pour fin Juillet 1990

III.3.2.2.1 Les objectifs de l'Etude Détaillée

L'objectif de l'étude détaillée est d'approfondir l'analyse effectuée lors de l'étude préalable. Cette étape permet de préciser certains points et de procéder aux choix finaux de notre conception. De plus, et en dernier lieu, elle nous oblige à réviser et à affiner le planning de réalisation élaboré précédemment.

III.3.2.2.2 Précisions et choix finaux

a) Mécanisme de contrôle

Nous avons dû mettre en évidence les mécanismes de reprise et de contrôle correspondants au "non OK" des différentes opérations.

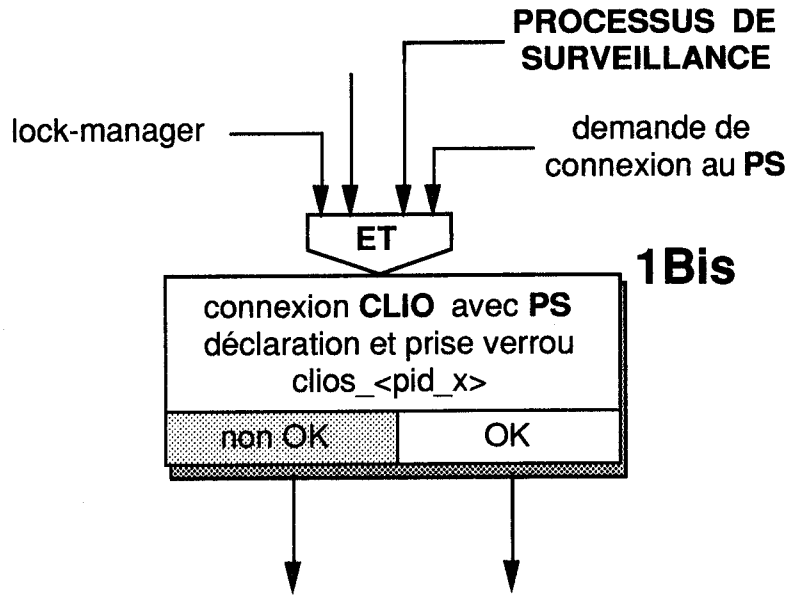


Fig III.3.2.2.2.a : exemple d'opération

Nous illustrons la réflexion menée sur chaque opération de nos modèles de traitement par un exemple, en reprenant l'opération de la figure précédente.

Il faut contrôler au moment de la connexion d'un utilisateur (opération 1bis) s'il n'existe pas de problèmes concernant les verrous, les mailboxes, les AST (déclaration, création, assignation). Il s'agit aussi de vérifier les comptes rendus de la communication avec le PS (message négatif de connexion émis par le PS (surnombre d'utilisateurs, mise à jour table PS impossible ...)).

b) Eléments techniques

Il nous faut aussi indiquer les derniers éléments techniques nécessaires: format de la table PS, format des messages de communication, libellés des différents messages d'erreurs, noms et algorithmes des différents modules et fonctions à réaliser.

On peut illustrer rapidement ces choix d'éléments techniques en commençant par le format de la table PS.

Pour pouvoir fonctionner, le mécanisme du PS requiert l'utilisation d'une "table" en mémoire, nécessaire à la détermination des différents utilisateurs CLIO présents. Il s'agit de définir toutes les informations qu'elle contient. La table peut évoluer au fur et à mesure de l'utilisation du PS. Nous décrivons dans la figure suivante le format de cette "TABLE PS" correspondant aux informations contenues minimales et nécessaires:

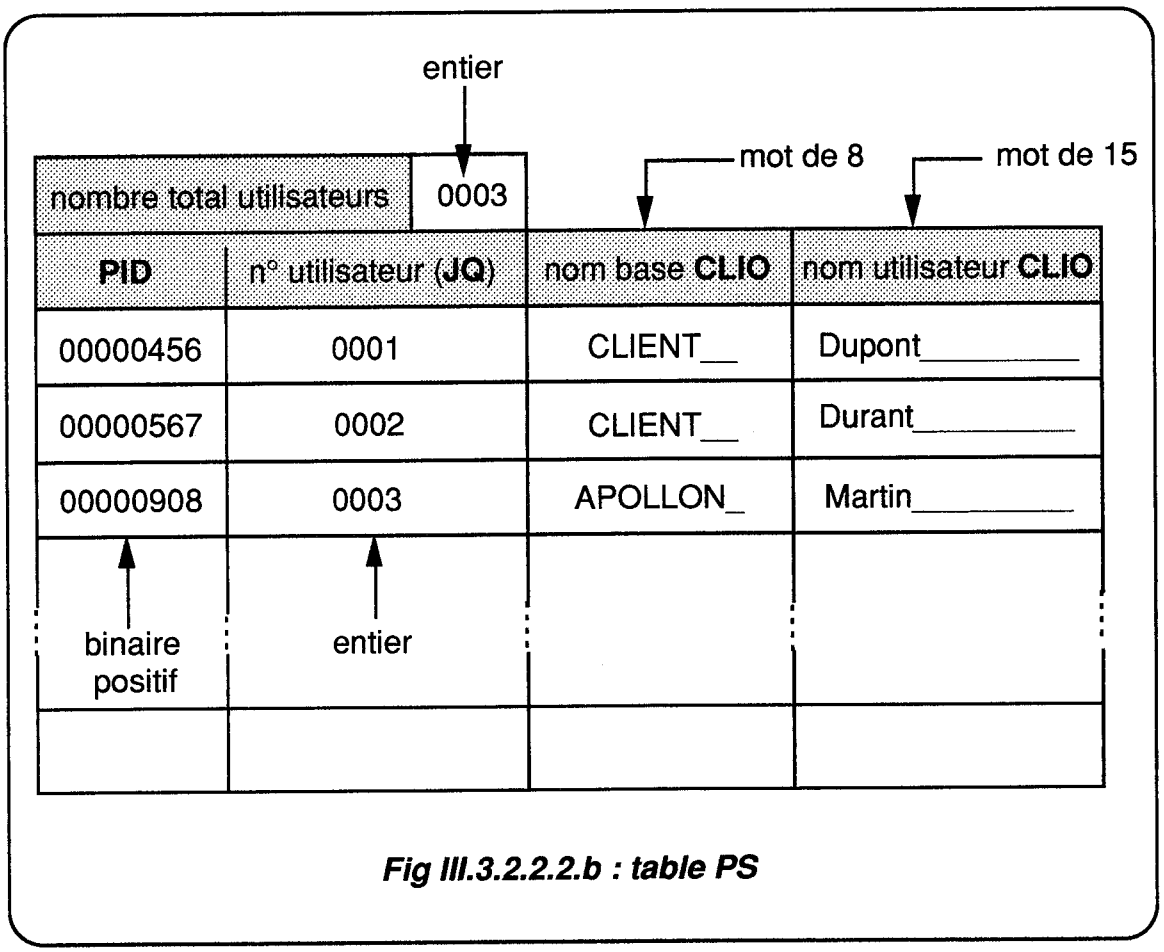


Fig III.3.2.2.2.b : table PS

* Le PID est indispensable pour identifier l'utilisateur
* Le numéro d'utilisateur CLIO est identique à celui de son JQ. Il est nécessaire au PS afin que celui-ci identifie le JQ et le restaure.
* Le nom de la base et le nom de l'utilisateur CLIO permettent de délivrer des informations sur l'utilisateur, de le contrôler et de cerner la base sur laquelle il se trouve.

Nous avons montré lors des chapitres précédents qu'il existe, entre le PS et les utilisateurs, une communication qui s'effectue à l'aide de mailboxes (boîtes aux lettres) CLIOPS_<machine> et CLIOS_<pid>. Nous présentons ces messages, la définition de leur format et la liste nécessaire à ce dialogue:

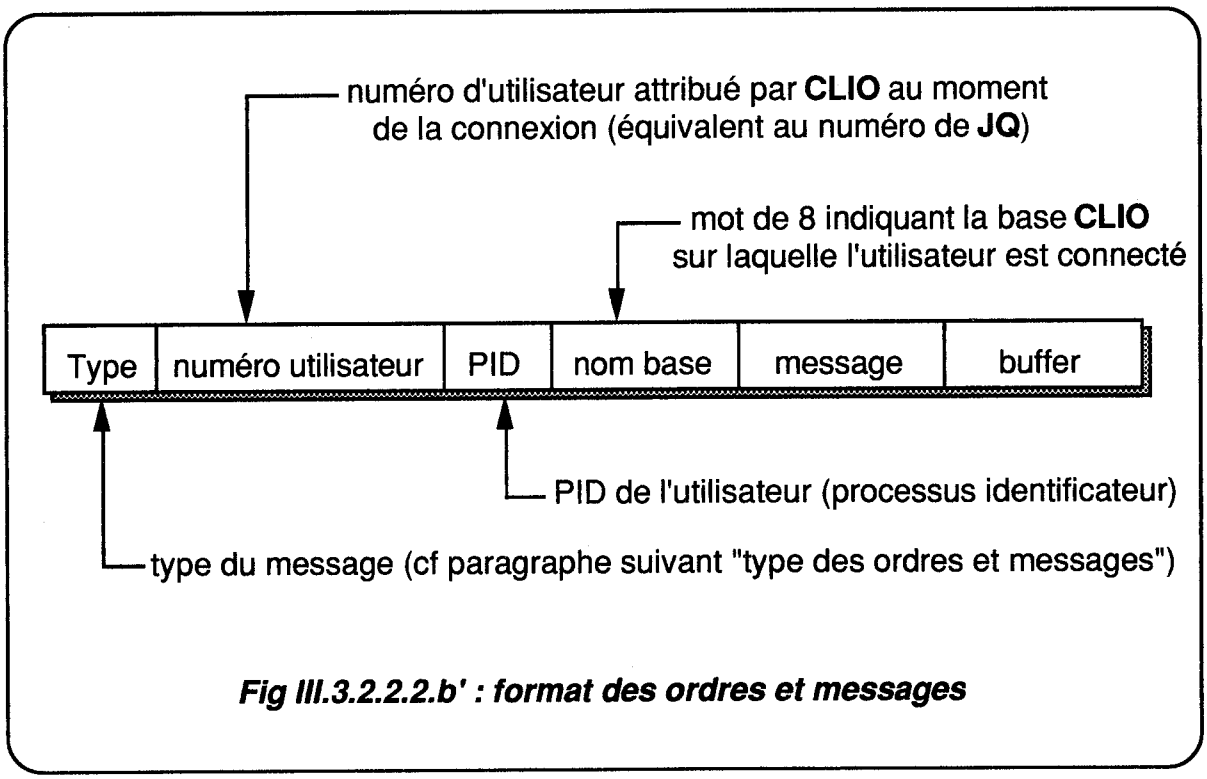


Fig III.3.2.2.2.b' : format des ordres et messages

* "Type" identifie le sens du message.
* "Numéro d'utilisateur", "PID", "nombase" caractérisent l'utilisateur.
* "Buffer" complète le champ "message".

Dans les paragraphes précédents, nous avons mis en évidence les différents types de messages qui circulent entre le PS et les utilisateurs via les mailboxes de communication. Ce paragraphe les précise et nous distinguons deux catégories de messages :

* la première appelée "ordre" est un message envoyé par le PS à un ou plusieurs, voire même à tous les utilisateurs. Il est donc déposé par le PS sur les mailboxes CLIOS_<pid> et lu ultérieurement par les utilisateurs, "réveillés", "activés" par l'arrivée de cet ordre (ast: ast_ordre).

* la deuxième catégorie intitulée "message" est un message envoyé par un utilisateur au PS. Il est déposé dans la mailbox CLIOPS du PS provoquant le "réveil" de celui-ci afin qu'il le lise (ast: ast_message) :

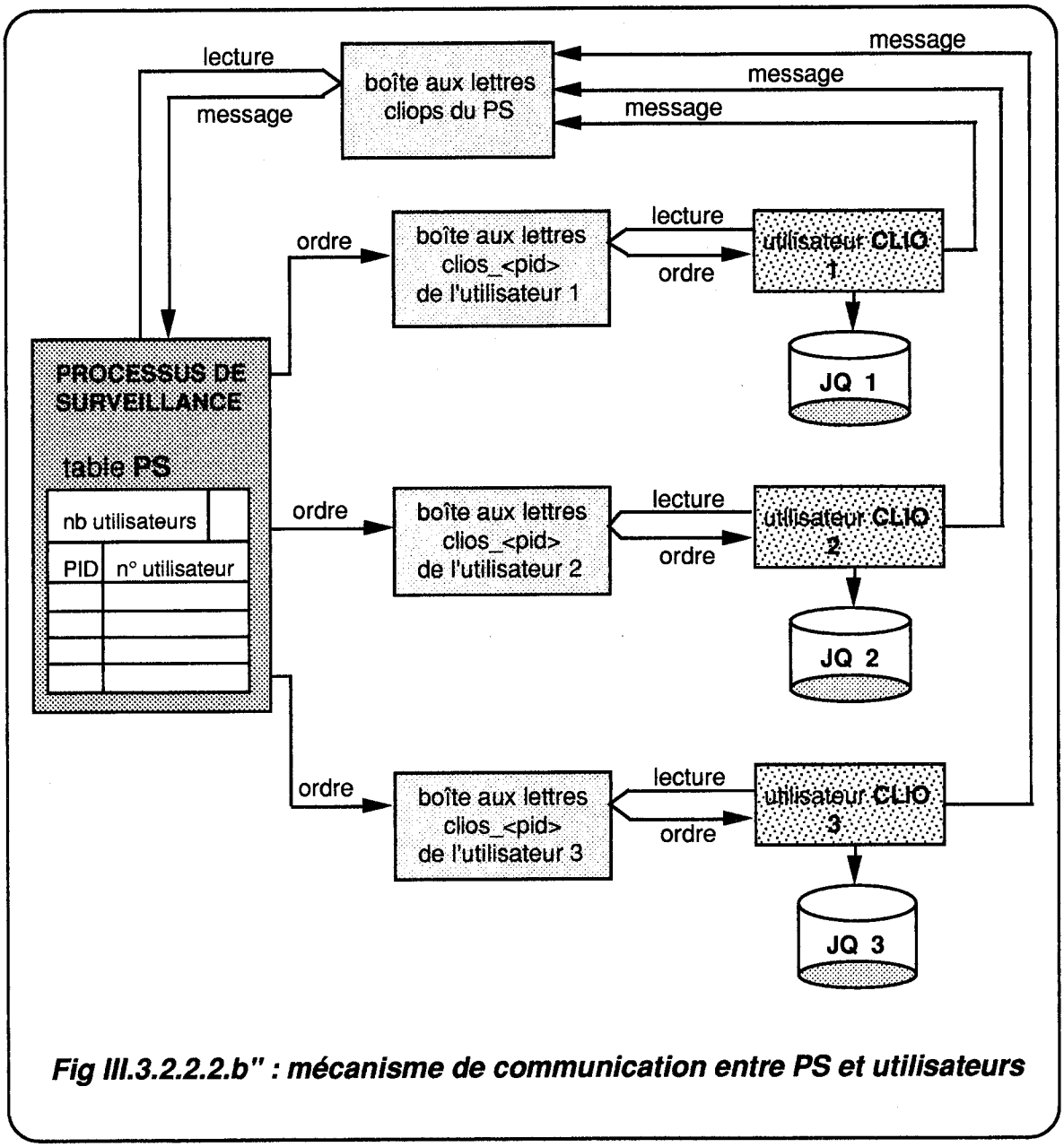


Fig III.3.2.2.2.b" : mécanisme de communication entre PS et utilisateurs

Dressons maintenant la liste des ordres et messages:

ordres
Processus ---------> utilisateur
de Surveillance

stop
rollback
redémarrage (ordre de redémarrage après restauration d'un utilisateur tombé pour les utilisateurs rollbackés)

demande_info
compte_rendu (connexion, déconnexion,maj table ps exemple: trop d'utilisateurs, problème de mailbox, l'information est dans buffer)

messages
Processus <---------- utilisateur
de Surveillance

connexion
déconnexion
connexion_base
déconnexion_base
envoi_info
rollback_OK
arrêt_déconnexion_sans_autorisation (suite erreur terminale)
ps_stop_uti

Il s'agit d'un dialogue élémentaire susceptible d'être enrichi si le PS augmente ses fonctionnalités.

La méthode à ce niveau de l'étude nous amène à définir les modules, fonctions, procédures et algorithmes:

Modules:
nous pouvons diviser nos procédures en deux groupes ou modules:
1er module : connexion, déconnexion du PS
2ème module : modification de CLIO pour intégrer le mécanisme du PS.
Leurs fonctions étant bien différentes, cette division nous semble nécessaire.

Fonctions, Tâches:
pour la dénomination des fonctions, nous nous sommes fixés la règle suivante:
<appelant>_<action>[_<destinataire>]

Ce qui implique comme fonctions:
1) PS_CONNEXION
2) PS_DECONNEXION
3) UTI_CONNEXION
4) UTI_DECONNNNEXION
5) PS_STOP_UTI
6) PS_RESTAURE_UTI-TOMBE
7) PS_MAJ-TABLE-PS
8) UTI_ROLLBACK
9) UTI-STOPPE_RESTAURATION

Il convient aussi de réaliser les divers AST:

AST déclarées par le Processus de Surveillance:

10) AST_message (déclarée au moment de la création de la mailbox cliops)
11) AST_chute_uti (déclarée au moment de la prise du verrou clios_<pid>)

AST déclarées par les utilisateurs:

12) AST_ordre (déclarée au moment de la création de la mailbox clios_<pid>)
13) AST_arrêt_PS (déclarée au moment de la prise du verrou cliops)

Algorithmes

A ce stade, nous avons aussi déterminé les algorithmes des programmes que nous ne livrerons pas ici car ils ne sont pas le propos de ce mémoire.

Erreurs

Nous avons également essayé de fixer le format des messages d'erreurs et la liste, car il convient, avant même la phase de réalisation, de pouvoir déterminer l'ajout de nouveaux messages et d'anticiper les problèmes d'intégration.

c) Détermination d'une méthode de suivi d'avancement de la réalisation:

Afin de superviser la réalisation du projet, il nous semble nécessaire de disposer de documents de suivi. La méthode Merise sensibilise au problème mais ne fournit pas de modèle. Aussi, nous avons étudié ce que les autres méthodes nous proposaient. Nous nous sommes inspirés d'une méthode déjà utilisée sur un projet pour sélectionner quatre documents de suivi : un rapport de situation de projet (RSP) qui permet de connaître l'état d'avancement, une fiche de tâche (FDT) qui correspond à une feuille périodique de réalisation de fonction, une fiche de rapport d'incident (FRI) qui permet de réajuster les prévisions de réalisation et enfin un document planning (PLA) (cf annexes).

d) Choix finaux

Nous illustrons nos choix finaux par deux exemples : le premier correspond à notre choix du langage de programmation, et le deuxième au choix de "qui doit activer CLIOPS ?"

Langage de programmation

Pour sa partie "moniteur", CLIO/VAX (non portable) est écrit principalement en Pascal. Une petite part de CLIO/VAX, difficilement programmable voire irréalisable en Pascal, est constituée en macro-assembleur.

Le "noyau" du générateur d'écran "GENICS" portable sur tous les CLIO ainsi que la partie communication ("station CLIO"...) sont en C mais là encore, cela ne correspond qu'à une infime partie du logiciel.

Notre choix s'est porté sur Pascal. Il nous semblait que cela nous permettrait d'augmenter notre productivité et d'économiser par la suite des efforts de maintenance. En effet, une grande partie de CLIO étant programmée en Pascal, il était possible de récupérer des primitives CLIO déjà écrites ainsi que des modèles d'appel aux primitives systèmes. Pascal est un langage modulaire permettant une programmation "propre", lisible et sûre. De plus, le compilateur sous VMS est très optimisé.

Activation de CLIOPS

La littérature informatique avec [AKOKA84] nous indique les douze fonctions de l'ABD (Administrateur de la Base de Données). Parmi ces fonctions, trois concernent la sécurité. [AKOKA84] explique <<l'ABD est responsable de la sécurité de la base, de l'établissement et du maintien d'un certain nombre de contrôles ...>>.

De plus, l'"ABD" CLIO effectue déjà des missions importantes d'administration de la base et de sécurité (outils d'arrêt des utilisateurs, activation du cache disque). Il convient donc de laisser les responsabilités de l'activation de CLIOPS à cet administrateur.

III.3.2.2.3 Planning (révision)

Notre étude détaillée prit fin avec la livraison d'un document fin septembre 1989, en totale conformité avec les prévisions du planning.

De plus, le planning nous semblait toujours valide, et nous étions à même d'apporter certaines précisions concernant la tâche 2 (réalisation et test). Nous pouvions fixer une date pour la sortie prévue de notre maquette: décembre 1989.

Nous avons aussi planifié deux nouvelles tâches 5 et 6, correspondant respectivement à la spécification détaillée et à la réalisation de la phase 2 de CLIOPS (version 2: CLIO en environnement cluster).

Il restait seulement une interrogation: sur quelle version de CLIO le processus de surveillance (CLIOPS) serait-il intégré ? Cette réponse pouvait influencer la date de livraison du produit final. Deux possibilités s'offraient aux responsables de CLIO: soit la version actuelle V4, soit la future version V6 (date de disponibilité proche du mois de mars donc du début de la phase d'intégration). A cette époque, le choix se portait plutôt sur CLIO V6.

Compte tenu de ces précisions, nous pouvions établir un nouveau planning:

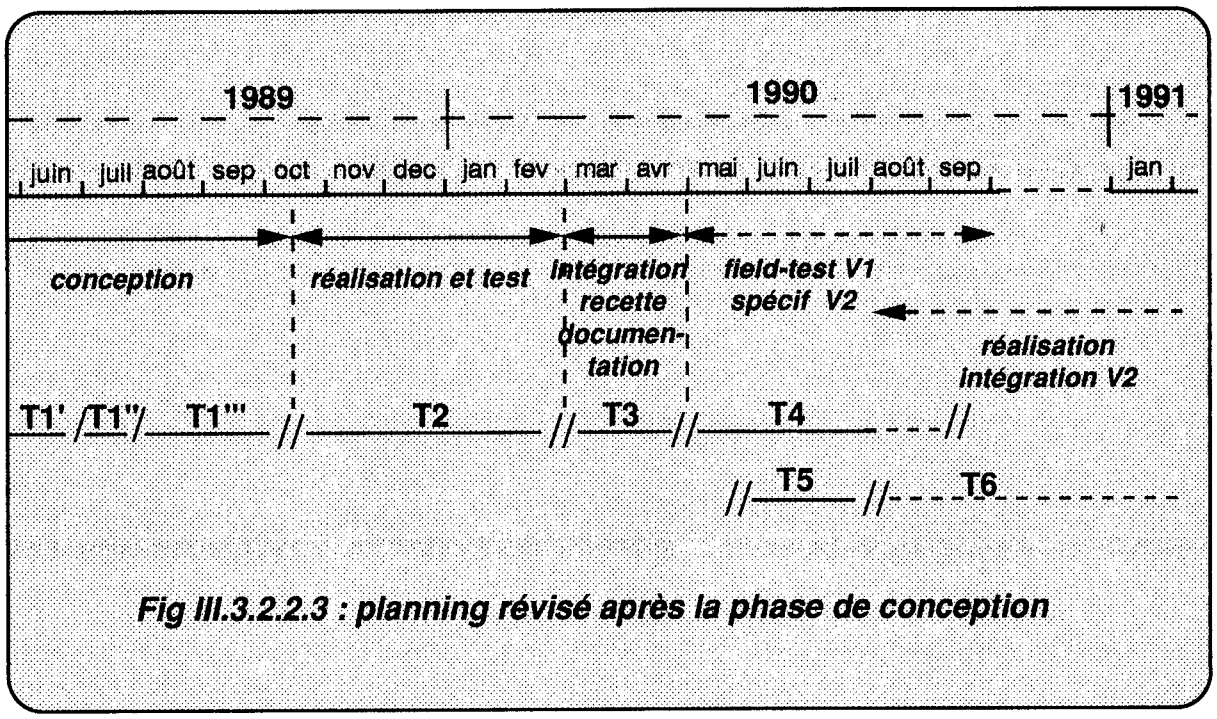


Fig III.3.2.2.3 : planning révisé après la phase de conception

III.3.3 L'ASPECT RELATIONNEL DE LA CONCEPTION:

III.3.3.1 L'interface concepteur/demandeur

Les demandeurs correspondent avant tout à des clients confrontés à des problèmes et émetteurs de désirs, mais aussi à la direction technique soucieuse de satisfaire ces besoins et d'anticiper les demandes futures.

Nous avons pu mesurer pendant la phase de recueil (cf.III.3.1.1.1) la nécessité d'un dialogue interactif avec les demandeurs afin de déterminer leurs besoins de façon synthétique et exhaustive. Cette phase est essentielle car le cahier des charges est l'outil de base du concepteur et toute définition imprécise ou mal formulée peut conduire à une traduction erronée du besoin. De plus, il est parfois primordial de s'adapter au langage de chaque interlocuteur afin de pouvoir retranscrire les problèmes des utilisateurs finaux.

III.3.3.2 L'interface concepteur/valideur

Il s'agit d'une des relations fondamentales du projet. En effet, la méthode demandait de soumettre chaque phase importante de l'analyse à une validation. Les membres de l'équipe CLIO/VAX, aidés de spécialistes, participèrent à cet exercice important. Ce fut l'occasion de goûter aux joies de la polémique édifiante. Il en a résulté une aide ponctuelle mais fondamentale, grâce aux critiques et aux visions différentes, rendant le travail de conception d'autant plus intéressant.

III.3.4 DIFFICULTES RENCONTREES PENDANT LA PHASE DE CONCEPTION, ENSEIGNEMENTS TIRES:

Je pense qu'il n'y a pas eu, pendant cette phase d'analyse, de difficultés autres que celles habituellement rencontrées. La partie conception constitue toujours un moment délicat, et la réalisation d'une étude exhaustive d'un problème se rapportant à la sécurité informatique, est ardue.

La difficulté me semble aussi résider dans le caractère parfois solitaire du travail, de la recherche d'une idée: une certaine "angoisse" devant la page blanche...

La méthode, dans sa démarche logique, donne la priorité à la recherche du "QUOI", puis du "QUI", "OU" ,"QUAND" pour terminer par le "COMMENT". Tout au long de l'analyse, il m'a semblé difficile d'éloigner de mon esprit le "COMMENT".

De plus, la méthode Merise n'apportant pas de formalisme pour le suivi de réalisation, il m'a donc fallu en déterminer un, ce qui a constitué un travail supplémentaire.

Mais la méthode, en tant que guide jalonnant mon travail et aide à la validation (par dialectique critique) a rendu cette phase plus aisée et plus attrayante.

III.4 REALISATION DE LA MAQUETTE

Aprés la phase de conception de CLIOPS, il nous semble intéressant d'évoquer l'expérience enrichissante que constitue le maquettage.

III.4.1 QU'EST-CE QU'UNE MAQUETTE ? DIFFERENCE AVEC UN PROTOTYPE

Avant d'analyser notre maquette, il est nécessaire d'éclaircir cette question afin de se fixer de justes objectifs.

La maquette:

La définition d'une maquette nous est donnée par [GENIEd]: <<la maquette permet de pallier à l'inconvénient de la constatation tardive. De plus, la maquette permet de valider que le système imaginé et l'idée du besoin concordent effectivement. Elle permet aussi d'examiner de façon détaillée et systématique les hypothèses de solution. La maquette constitue une simulation.>> [GENIEc] résume: <<la maquette permet de vérifier que le système est réalisable et correspond bien à ce que l'on attend>>. Pour [GENIEb]: <<elle a pour objectif de vérifier que les spécifications sont conformes aux besoins réels>>, <<cet objectif doit être atteint rapidement sans se préoccuper de certaines contraintes propres au développement (portabilité, évolutibilité,...); une solution consiste en l'élaboration de maquette jetable>>.

Le prototype:

[GENIEa] nous apporte une définition: <<le mot prototype, qui signifie étymologiquement "premier de série", se rapporte à un premier exemplaire, qui possède les mêmes fonctionnalités que l'objet final et qui sert à des tests en contexte réel>>. [GENIEc] précise: <<le prototype d'un produit est un modèle qui sacrifie à la précision dans certains domaines pour permettre une vérification rapide des fonctions du produit. Un prototype est donc une forme de maquette exécutable>>, et <<une réalisation conforme à la maquette>> mais <<le prototype, une fois validé, doit pouvoir être "récupéré">>[GENIEb].

Nous conclurons sur cette précision de [GENIEd] : <<on peut considérer le prototype comme une réalisation rapide qui va à l'essentiel des comportements significatifs ou, au contraire, détaille certains points précis qu'il faut éclairer au détriment des autres.>>

III.4.2 NOS OBJECTIFS

A travers cette maquette, nous voulons avant tout valider les aspects techniques, les mécanismes (le "comment"), démontrer par cette simulation que la solution est réalisable (faisabilité) et enfin saisir l'aspect final du produit. Il s'agit de confirmer notre analyse.

III.4.3 LES RESULTATS

Le PS est entièrement programmé dans sa partie fonctionnement. Seules les actions de restauration ne sont pas implémentées. Le CLIO est quant à lui totalement simulé: pas de vraie connexion à CLIO, uniquement la connexion au PS et un dialogue avec celui-ci.

Nous pouvons répondre aux deux questions qui nous préoccupent:
- Le PS détecte bien tous les arrêts anormaux des utilisateurs
- Le dialogue PS (CLIOPS) utilisateur (CLIO) est totalement réalisable.

De plus, il nous est possible de mettre à l'épreuve notre première batterie de tests (arrêt d'éléments du réseau) et en évidence la nécessité de tests supplémentaires. Nous pouvons aussi, par là-même, effectuer nos premières mesures de performances (consommation CPU du PS, temps de réponse ....).

III.4.4 LES DIFFICULTES RENCONTREES ET LES ENSEIGNEMENTS TIRES:

Les difficultés les plus importantes sont liées à la programmation de la maquette. Malgré le choix du langage PASCAL, il restait tout de même beaucoup de problèmes de mises au point des appels systèmes (paramétrages, choix...). En effet, un certain nombre de primitives n'avaient jamais été utilisées dans les développements antérieurs de CLIO. De plus, le stage Digital "synchronisation et communication entre process", n'a pu être planifié plus tôt et n'a été effectué que pour la phase d'intégration.

Il n'a pas été aisé d'effectuer les tests multi-utilisateurs. Ces derniers ont mis en évidence les problèmes de chutes simultanées nécessitant une deuxième validation de notre maquette. L'utilisation de "traces" dans des procédures asynchrones et le debooger ont généré des effets de bord, parfois difficiles à cerner.

Nous tirons différents enseignements de ce maquettage. Il s'agit en somme de prises de conscience:

- des avantages de la maquette: utilité primordiale de celle-ci afin de valider les spécifications (principe de fonctionnement) et d'envisager l'avenir plus directement (performance, test..). De plus, cette maquette, par son caractère concret, est source de remotivations personnelles.
- des limites de la maquette: son utilisation est restreinte, il ne s'agit pas de la réalisation définitive
- de nos propres limites ...: la réalisation de cette maquette requiert de solides compétences, aussi nous met-elle face à nos manques et à nos désirs de formation.

III.5 INTEGRATION DE CLIOPS A CLIO

Elle correspond à la dernière phase de la version 1 de CLIOPS. Nous ne détaillons pas ici cette intégration, mais nous en présentons les objectifs, et un point qui nous paraît délicat: les tests et la validation du produit.

III.5.1 NOS OBJECTIFS

Il s'agit d'aboutir à un produit final livrable en clientèle. Il convient donc à partir de la maquette et des tests d'intégration effectués lors de l'étape précédente, d'intégrer au sein de CLIO le nouveau mécanisme. La version support de l'intégration n'est pas CLIO V6, comme le prévoyait la première hypothèse , mais CLIO V4 -version stabilisée livrée en clientèle depuis JUIN 1989-.

Nous avons décidé de suivre la méthode classique d'élaboration d'une nouvelle version CLIO, méthode ayant fait depuis longtemps ses preuves. Elle doit suivre trois étapes. Seule la dernière a dû être adaptée:

- élaboration d'un prototype (les objectifs sont classiques et présentés dans le paragraphe précédent)
- élaboration d'une version field-test validée (alpha et béta tests) diffusée à certains clients
- la dernière phase correspond habituellement à la diffusion générale.
  CLIO/VAX, muni de la version 1 de CLIOPS, ne suivra pas cette règle, mais servira de point de départ à la version 2 (PS sur cluster) qui fera ultérieurement (1991) l'objet d'une livraison à tout le parc des clients.

III.5.2 LES TESTS, LA VALIDATION DU PRODUIT

Notre objectif de diffusion en clientèle nous oblige bien évidemment à insister sur un point qui nous paraît fondamental: les tests et la validation du produit. Ils sont déterminants de la qualité du produit mais complexes à effectuer. L'équipe et SYSECA possèdent une longue expérience en matière d'élaboration de nouvelles versions de logiciels en particulier CLIO mais aussi GENICS, GAIA ...

Une validation standard de CLIO est effectuée sur le produit avant chaque livraison d'une nouvelle version (travail automatique de plus d'un mois). Il convient d'enrichir cette validation de procédures testant chaque nouvel outil ou fonctionnalité ajoutés à CLIO.

Reprenant cette démarche classique, nous avons perfectionné et intégré les tests de la maquette. La figure III.3.1.1.1.1 ("exemple de réseau Digital") illustre le type d'éléments que nous avons mis en défaut (pannes) pour tester et valider notre nouvelle sécurité.

III.5.3 L'ASPECT RELATIONNEL DE L'INTEGRATION : L'INTERFACE CONCEPTEUR / REALISATEUR

Je ne fus pas seul durant la phase de réalisation: une stagiaire et un ingénieur de l'équipe CLIO/VAX ont participé pleinement à la tâche de programmation et d'intégration sous contrainte des éléments fondamentaux de l'analyse. D'autres apportèrent leur contribution passagère mais fondamentale.

La qualité des réalisateurs modifia quelque peu les rapports habituels qui peuvent exister entre concepteur et réalisateur: ce fut une réelle collaboration, tant leurs apports, compétences et implication furent importants.

III.5.4 LES DIFFICULTES RENCONTREES ET LES ENSEIGNEMENTS TIRES:

L'analyse a certainement quelque peu négligé les problèmes liés aux différents modes de fonctionnement de CLIO. Aussi, l'intégration à l'existant nous confronta à certaines difficultés supplémentaires: le fonctionnement en mode "interface langage" (CLIO enchassé dans un programme écrit par exemple en PASCAL ou en COBOL) nécessite un dialogue différent du CLIO L4G (ex: la déconnexion de CLIO est implicite ainsi que le changement de base).

Outre ces difficultés techniques, le travail d'intégration s'avère aussi plus long et plus fastidieux que prévu: il nécessite non seulement une excellente connaissance de CLIO afin de déterminer les points "d'ancrage" (appel) du nouveau mécanisme (connexion au PS, déconnexion au PS ...), mais aussi une excellente pratique du langage Pascal.

Enfin, le travail de mise au point (d'élaboration) des tests et de la validation fut important, malaisé, requérant méthode et patience ainsi qu'un matériel conséquent (decserveur, delni, ...) apte à feindre les différentes pannes.

Donc, nous avons pris conscience qu'il convient de ne pas négliger la phase de réalisation (d'intégration) aux niveaux temps, ressources humaines et matériels car la rigueur et les moyens déterminent la qualité du produit final. Concrètement, il est nécessaire par exemple de réserver et de planifier les temps machine, les plages horaires de test d'arrêt machine etc...

CHAPITRE IV

QUEL PROCESSUS DE SURVEILLANCE POUR DEMAIN ?

QUELLE SECURITE POUR DEMAIN ?

IV.1 LE CLIOPS DE DEMAIN : CLIOPS VERSION 2

Le CLIOPS version 2, que nous devons maintenant spécifier et réaliser sera disponible en 1991 comme l'illustre la figure suivante. Il assurera le fonctionnement total de CLIO en environnement Cluster (grappe d'ordinateurs, multi-machines) et le dialogue avec de nouveaux outils. (cf fig.IV.1.b et IV.1.c)

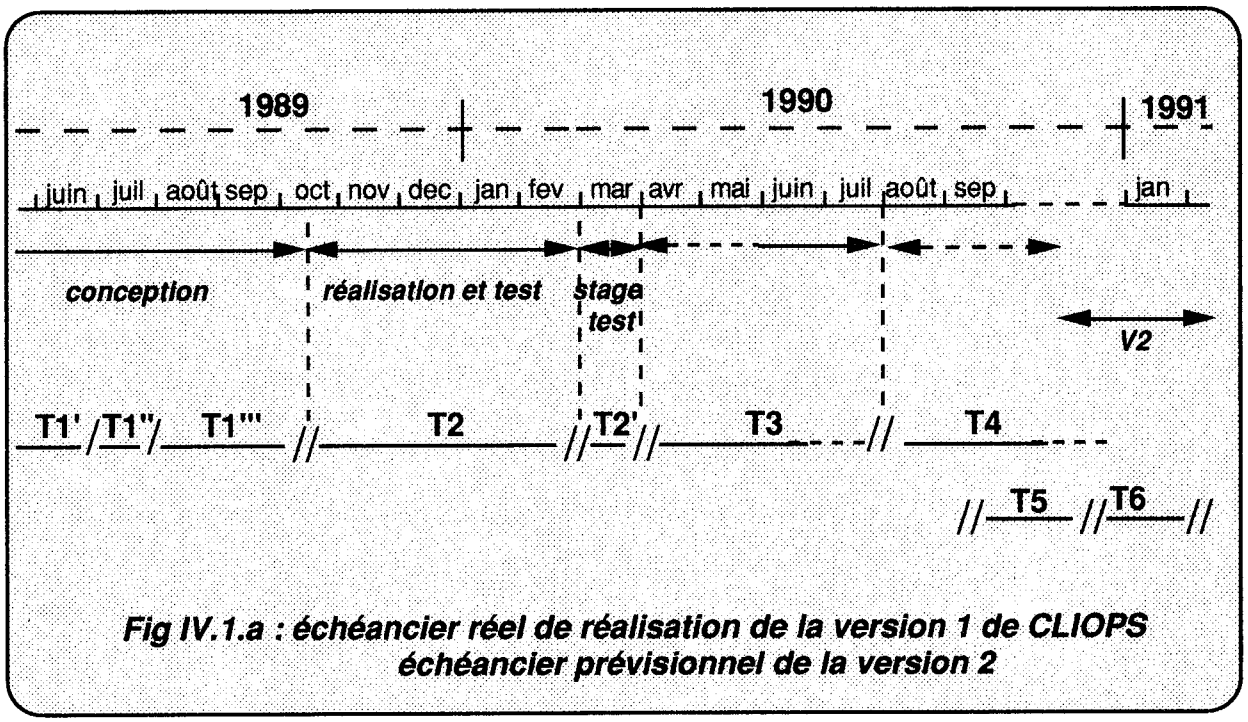


Fig IV.1.a : échéancier réel de réalisation de la version 1 de CLIOPS
échéancier prévisionnel de la version 2

Le CLIOPS version 2 utilisera les mêmes principes et mécanismes que ceux de la version 1, mais il sera constitué d'un PS par machine, élément du cluster. Ces différents surveilleurs dialogueront entre eux afin de se tenir informés des diverses connexions et déconnexions intervenant sur leur ordinateur. Chaque PS aura la charge des incidents survenant sur la machine qu'il surveille. Par contre, lors de l'arrêt anormal d'un élément du cluster, le "lockmanager" alertera (libération verrou cliops_<machine>) les PS des autres noeuds. L'un deux effectuera la restauration des utilisateurs "tombés".

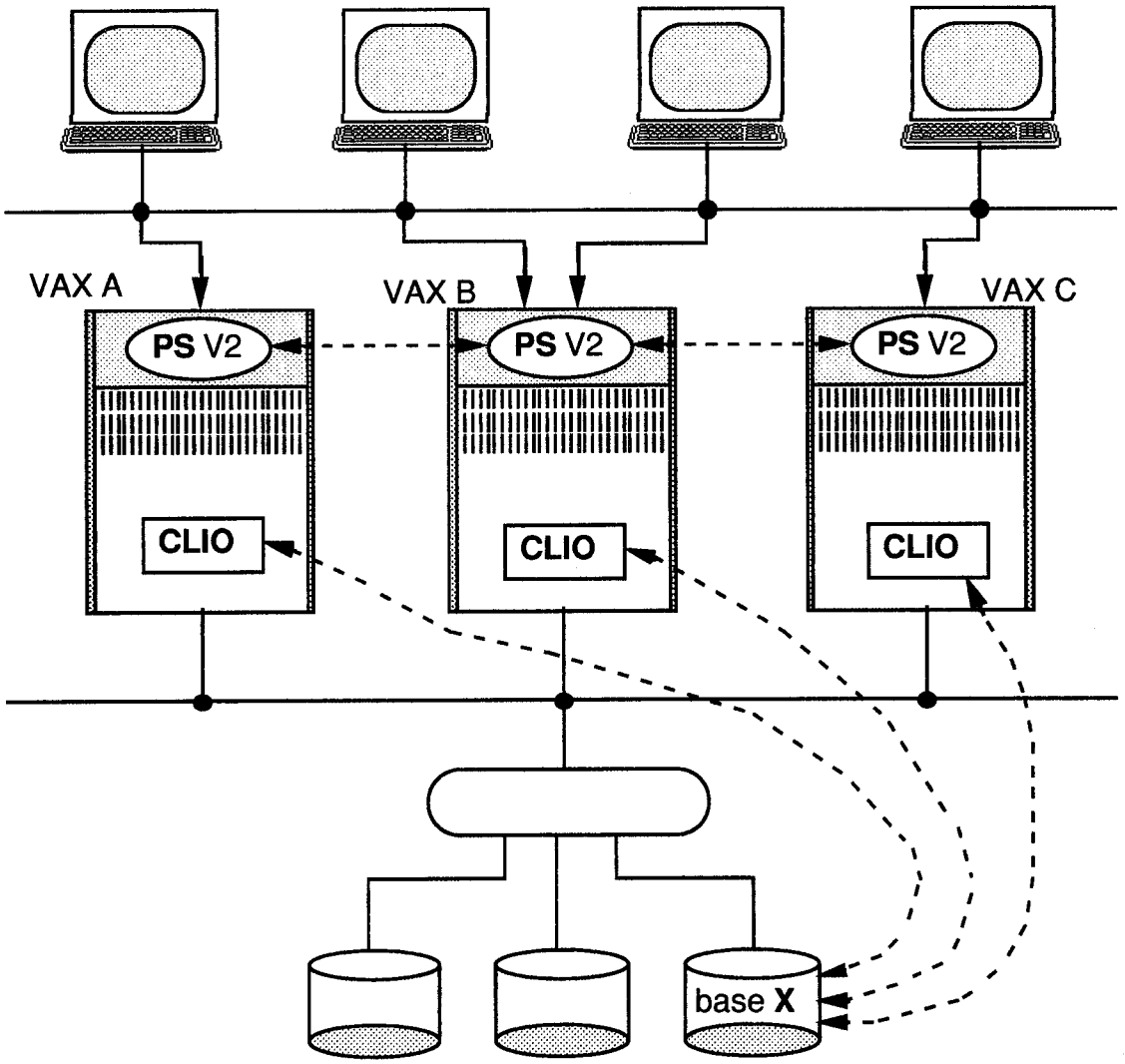


Fig IV.1.b : CLIO multi machines - utilisateurs multi machines

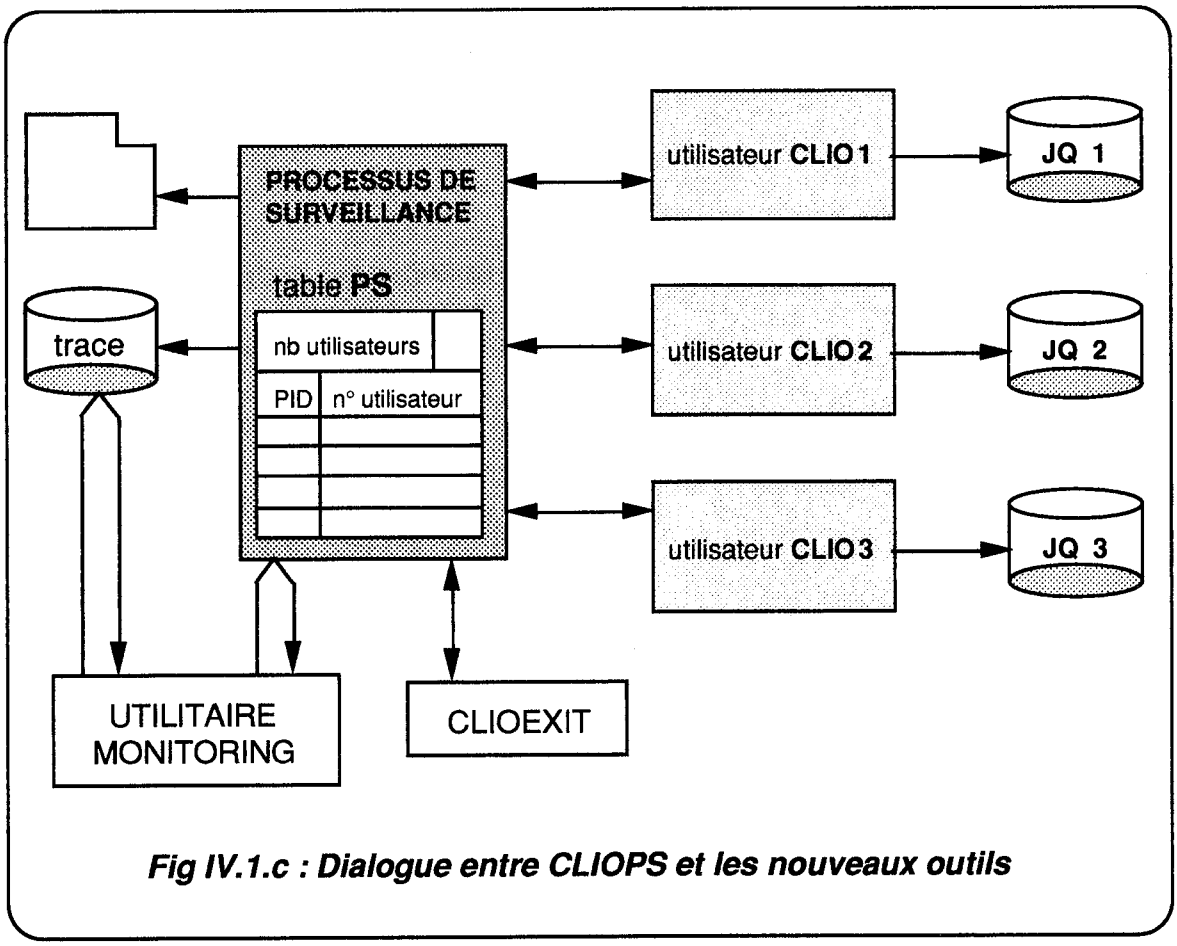


Fig IV.1.c : Dialogue entre CLIOPS et les nouveaux outils

IV.2 LE PS DE DEMAIN

Il convient de terminer ce mémoire en envisageant l'avenir, car il est vrai que l'informaticien est toujours tourné vers le futur dans son désir d'anticiper, de prévoir.

Nous notions dans le paragraphe II.2, que les SGBD du futur seraient les SGBDOO (Orientés Objets), les SGBDR (répartis), ainsi que les SGBDD (déductifs) . C'est ce que nous confirme [GARVAL90] dans son ouvrage récent intitulé: "SGBD avancés: Bases de données objets, déductives, réparties".

Il ressort que la sécurité du futur sera davantage fonction des SGBD de demain, (les SGBDOO et SGBDR) que de l'émergence d'un nouveau concept de sécurité.

Mais d'une part, l'évolution vers les SGBDOO pose le problème de savoir comment doit évoluer la sécurité compte tenu de la manipulation de "gros objets" impossibles à sauvegarder dans un journal de sécurité.

D'autre part, l'évolution vers les SGBDR, amène deux nouvelles difficultés: tout d'abord les besoins de normalisation, qui impliquent que la sécurité doit répondre à des questions du type: "qui doit rollbacker sur deadlock ?" etc ... puis les problèmes de sécurité liés à la répartition des systèmes reliés entre eux par des réseaux.

[GARVAL90] précise: <<la fiabilité d'une base de données répartie peut s'obtenir en adaptant quelque peu les techniques de journalisation et points de reprise des systèmes centralisés.>> Il introduit aussi la notion de Coordinateur avec la validation en deux étapes des transactions réparties. Le Coordinateur est capable de prendre une décision globale (validation ou abandon de la transaction) compte tenu de la réponse des participants: "prêt", "non-prêt" signalant l'impossibilité de valider (exemple: panne locale, verrou mortel):

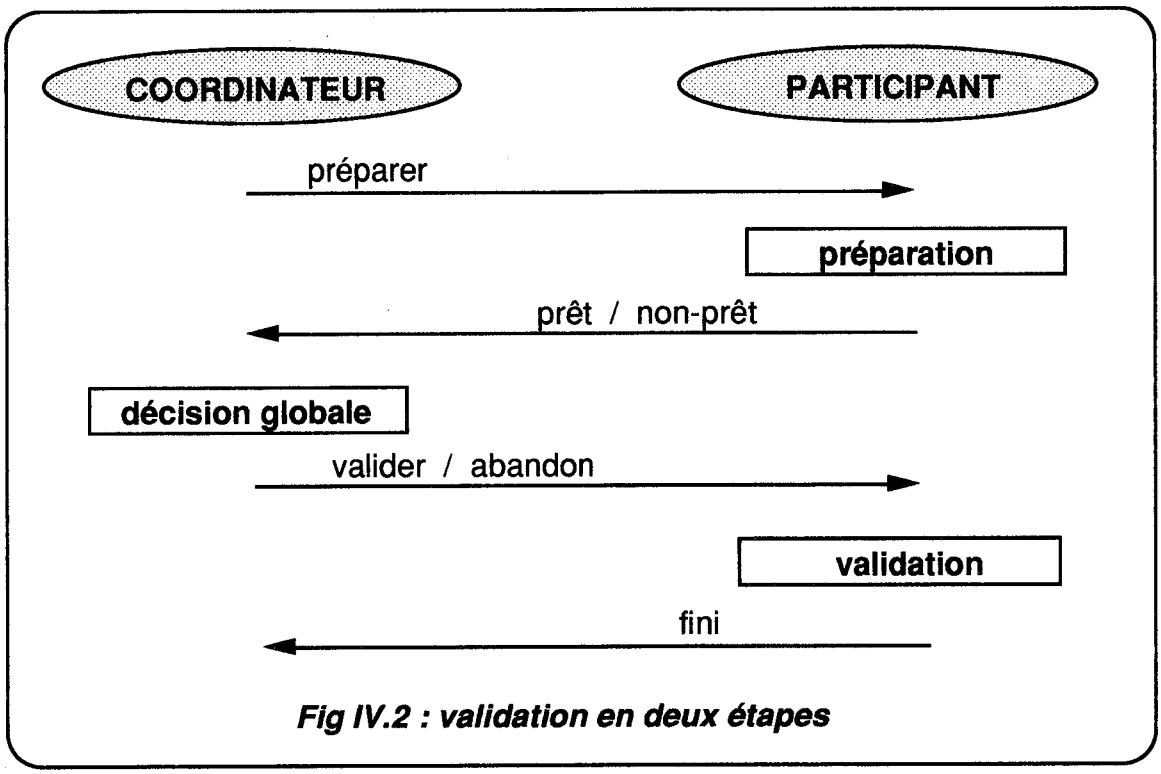


Fig IV.2 : validation en deux étapes

CLIOPS ne pourrait-il pas s'assimiler aux premières ébauches du Coordinateur de l'avenir ?

CONCLUSION

Les premières conclusions concernent la conception, la réalisation et les résultats obtenus:

Les objectifs semblent atteints: la réalisation d'une version de CLIO/VAX, munie du nouveau mécanisme de sécurité que nous avons conçu et dont l'élément moteur est un Processus de Surveillance (CLIOPS), assure bien à CLIO une sécurité totale et particulièrement dans un environnement réseau défaillant (panne). En effet, il détecte toute fin anormale d'un utilisateur CLIO et met en oeuvre sa restauration.

Les contraintes semblent elles aussi respectées: nous avons obtenu un logiciel fiable dont les tests de performances n'ont montré qu'un coût modéré pour l'ajout de cette nouvelle sécurité. Ce mécanisme s'intègre à CLIO sans une refonte profonde de celui-ci. Il est, de plus, facile de mise en oeuvre. CLIOPS (version 1) est évolutif: il s'agit d'un système central sachant dialoguer avec les utilisateurs ou l'administrateur, capable de retransmettre des ordres. Il est possible de lui greffer de nouvelles fonctionnalités par addition d'outils aptes à communiquer avec lui.

La première version de CLIOPS a donc atteint les objectifs fixés tout en respectant au mieux les contraintes. Il ouvre ainsi la porte vers la deuxième phase de notre projet, qui doit aboutir à une version commercialisable début 1991. CLIO/VAX sera alors pleinement utilisable sur Cluster (grappe d'ordinateurs) et, nous l'espérons, satisfera nos clients, juges eux-aussi de cette longue réalisation industrielle.

Ce travail sur CLIOPS fut aussi l'occasion pour nous d'un très grand enrichissement personnel. Ce projet nous apporta bien sûr de nouvelles compétences techniques mais aussi une importante expérience relationnelle: En effet, CLIOPS, par sa dimension, nous a permis de traiter un projet dans son intégralité, du recueil des besoins jusqu'à la réalisation industrielle .

La recherche bibliographique et l'étude de la concurrence furent l'occasion de beaucoup d'apprentissages sur les SGBD et leurs sécurités.

La conception du projet par son analyse suivant la méthode Merise, nous a enseigné l'envergure de cette étape mais aussi la délicatesse de ce travail parfois solitaire. Ainsi, nous avons pu mesurer l'importance de l'utilisation d'une méthode: d'autres que Merise auraient certainement convenu mais elles auraient nécessité plus d'investissements en temps de notre part. Merise a jalonné notre travail, apporté de la rigueur et permis de rendre la conception plus aisée et encore plus attrayante. Cependant, nous n'avons pas, et c'est là l'occasion de faire notre mea culpa, suivi la méthode dans toute son intégralité et ce pour des raisons d'ampleurs de projet et de planning. Nous aurions pu alors optimiser davantage les autres étapes et surtout relever dès le début les éventuelles imprécisions et ambiguïtés de notre cahier des charges.

Les réalisations de la maquette et du produit nous ont fait prendre conscience de la nécessité d'un maquettage qui valide l'analyse, et nous remotive par son caractère concret. Nous désirions utiliser des "feuilles de suivi": ce fut là une expérience peu concluante et vite abandonnée car la dimension du projet ne justifiait certainement pas leur emploi. On peut éventuellement le regretter, il aurait probablement fallu persister...

Ce travail de réalisation nous a apporté des connaissances approfondies sur le langage Pascal, sur le système d'exploitation VMS (notamment la gestion des processus et le fonctionnement de la communication inter process), ainsi que sur les éléments qui composent un réseau d'ordinateurs. Nous avons mesuré l'importance de la phase de réalisation qu'il convient de ne pas négliger du point de vue temps, ressources humaines et matérielles, car la rigueur et les moyens déterminent, avec le respect d'une bonne analyse, la qualité du produit final. De plus, nous avons découvert avec CLIOPS la complexité que constituent les tests et la mise au point de programmes incluant des communications.

Enfin, ce projet dans sa conception et sa réalisation, fut l'occasion de saisir toute la difficulté à élaborer et respecter un planning.

Il permit de mieux appréhender les relations humaines, dans leur richesse et dans leur diversité. Nous avons pu mesurer la nécessité du dialogue avec les demandeurs-clients, les joies de la polémique édifiante avec les collaborateurs, ainsi que l'importance de la confiance et du soutien de la direction.

Fort de cette expérience, je pense que si c'était à refaire, je suivrais le même chemin à certaines variantes près: étant donné l'extrême complexité du sujet, je serais probablement plus vigilant sur l'utilisation de la méthode, le respect des délais (planning), l'acquisition des compétences nécessaires aux niveaux technique et matériel. J'essaierais de toute évidence de mettre en application les choix désormais analysés comme les meilleurs. Je pense que tout de même nous ne nous sommes ni trop mépris ni égarés, et cela certainement grâce à l'emploi d'une méthode guidant notre chemin, mais aussi grâce à la qualité des personnes ayant participé à ce projet.

Enfin, ce mémoire fut l'occasion d'effectuer un bilan sur un travail qui nous tenait à coeur, et de livrer par écrit une expérience qui, si je devais la résumer en un mot serait sans hésitation définie par l'adjectif "riche".

A N N E X E S

	DATE __/__/__	PROCESSUS DE SURVEILLANCE	FDT	

F I C H E D E T A C H E (fonction)

REDACTEUR :	TACHE N° :

	DEBUT	FIN	H/J
DATES PREVUES			
DATES REVISEES			

DENOMINATION DESCRIPTION : _____

_____

OBSERVATION : _____

_____

_____

TYPE RECETTE INTERNE : EXTERNE : DATE : __/__/__

NOM DU RECETTEUR : _____

	DATE __/__/__	**PROCESSUS DE SURVEILLANCE**	FRI		

FICHE DE RAPORT D'INCIDENT

REDACTEUR :

FRI N° : ____

DETAILS INCIDENT : _____

_____

_____

_____

CONSEQUENCES : SURCROIT DE CHARGE ☐ RETARD PREVISIBLE ☐

INCIDENCE SUR LES TACHES N° ☐

☐

☐

INCIDENT INTERNE EXTERNE PORTANT SUR :

PERSONNEL

MATERIEL

LOCAUX

DOCUMENTS

	DATE __/__/__	PROCESSUS DE SURVEILLANCE	RSP	

RAPORT DE SITUATION DE PROJET

REDACTEUR :

N° TACHE fonction	DESIGNATION	DUREE estimée	/ / DP	PF	/ / DP	PF	/ / DP	PF	/ / DP	PF	/ / DP	PF	/ / DP	PF
	TOTAUX A REPORTER													

	DATE __/__/__	PROCESSUS DE SURVEILLANCE	PLA		

P L A N N I N G

REDACTEUR :

COLLABORATEURS TACHES (fonctions)	type travail	nombre unités								

B I B L I O G R A P H I E

OUVRAGES

[ABDE90] ABDELLATIF A., LE BIHAN J., LIMAME M. : *Oracle, les SGBD relationnels*, Edition Eyrolles, 1990, 2ème édition.

[ADIB83] ADIBA M., DELOBEL C. : *Bases de données et systèmes relationnels*, Dunod Informatique,1983.

[AKOKA84] AKOKA J. : *Les Systèmes de gestion de bases de données : Théorie et pratique*, Edition Eyrolles, 1984.

[COHEN89] COHEN J. : *Les SGBD Relationnels sous Unix, étude comparative*, Cabinet conseil Jo Cohen, janvier 1989.

[FLOBOU86] FLORY A., BOUZEGHOUD M. : *Bases de données relationnelles: mythes et réalités*, Actes des journées AFCET, 14-15 mai 1986 Paris, Editions Eyrolles, 1986.

[GALACSI89] Collectif: *Conception de bases de données, du schéma conceptuel aux schémas physiques*, Dunod, 1989.

[GARD83] GARDARIN G. : *Bases de données : les systèmes et leurs langages*, Editions Eyrolles, 1983.

[GARVAL85] GARDARIN G., VALDURIEZ P.: *Bases de données relationnelles: analyse et comparaison des systèmes*, Editions Eyrolles, 1985.

[GARVAL90] GARDARIN G., VALDURIEZ P.: *SGBD avancés, Bases de données objets, déductives, réparties*, Editions Eyrolles, 1990.

[GARVIA87] GARDARIN G., VIALLET F.: *Bases de données relationnelles, supra de cimcom*, Editions Eyrolles, 1987.

[GRAY78] GRAY J.N : *Notes on database operating systems, in Operating Systems : An advanced course*, édité par Goos and Houtmanis, Springer - Verlay.

[MIRA77] MIRANDA S : *Data security in centralized and distributed data bases*, Master thèse, université de Californie à Los Angeles, UCLA, 1977.

[MIRA86] MIRANDA S., BUSTA JM. : *L'art des bases de données : introduction aux bases de données*, tome 1, Edition Eyrolles, 1986.

[MIRA90] MIRANDA S., BUSTA JM. : *L'art des bases de données : les bases de données relationnelles*, tome 2, Edition Eyrolles, 1990.

[OLIVAR86] OLIVARES J. *: Traitement logique de l'intégrité et de l'organisation sémantique des connaissances dans les systèmes de gestion de bases de données*, Thèse de Docteur de l'Institut National Polytechnique de Grenoble, soutenue le 19 Juin 1986.

[PASL90] PASLEAU S. : *Apprendre Oracle*, Edition Nathan, 1990.

[PHI86] PHILIPPE S. : *Introduction aux bases de données*, Editest, 1986.

[PUJO85] PUJOLL G., SERET D., DROMARD D., HORLAIT E. : *Réseaux et télématique*, Edition Eyrolles, 1985.

[TARNIPA84] TARDIEU H., NANCI D., PASCOT D. : *Conception d'un système d'information*, les éditions d'organisation, 1984.

[TAROCO89] TARDIEU H., ROCHFELD A., COLLETTI R., PANET G.: *La méthode Merise: "Principes et outils"*, tome 1, les éditions d'organisation, 1989.

[TAROCO83] TARDIEU H., ROCHFELD A., COLLETTI R., PANET G.: *La méthode Merise: "Démarches et pratiques"*, tome 2, les éditions d'organisation,1986.

[ROCHMO89] ROCHFELD A., MOREJON J. : *La méthode Merise: "Gamme opératoire"*, Tome 3, les éditions d'organisation, 1989.

ARTICLES, RAPPORTS, MEMOIRES

[01INFO88a] *Un discours sur les méthodes*, 01 informatique n°1026, 19 septembre 1988.

[01INFO89a] *Les 500 de 01*, 01 informatique N°4 hors série, septembre 1989.

[01INFO89c] *Les Vingt de 01*, 01 informatique N°1068 , 10 juillet 1989.

[DICSI90] *Dossier d'information sur les constructeurs et services en informatique,* 10ème Année 1990, INDE (innovation et développement) 1990.

[DIGITALa] *Digital et les réseaux,* présentation du département Marketing promotion/Service Marketing Communication, Digital 1988.

[DIGITALb] *Quoi de neuf dans VMS V5.4 ?*, La lettre de High-Dec, 27 Février 1990.

[DIGITALc] *Decnet : The success continues !*, Flash Partenaires, Digital N°49, 10 avril 1990.

[GENIEa] CHOPPY C. : *Technique et aspect du prototypage*, Revue Génie Informatique n°3.

[GENIEb] MUENIER M. : *Le point de vue d'un industriel: Maquettage de définition et Prototypage de conception,* Revue Génie Logiciel n°3.

[GENIEc] MARTIN G.A : *APL et le Prototypage*, Revue Génie Logiciel N°3.

[GENIEd] ROAN A., TROY R. : *Maquettage de spécification de systèmes temps réel industriels par analyse structurée et automates*, Revue Génie Logiciel n°3.

[JAPS90] JAQUIN P., PETIT P., SAMSON F. : *Suivi de clientèle pour le SGBD CLIO*, Dossier de Méthodologie d'informatisation, CNAM CUEFA de Grenoble, Grenoble 1990.

[MBD89] SIBERTIN-BLANC C. : *Le modèle de données objet comme formalisme de modélisation d'une base de données*, La revue MBD-synthèse n°9, juin 1989.

[MINI89] *Bases de données de la boîte à fiches au SGBD orienté objet*, MINI & MICROS N°331/27, novembre 1989.

[SIL89] SILVESTRE D. : *Réalisation d'un serveur de communication OSI dans un réseau local Ethernet*, Mémoire CNAM Grenoble 1988.

DOCUMENTATIONS TECHNIQUES

[CLIOa] *Présentation de CLIO*, Syséca 1989.

[CLIOb] *Internes CLIO/VAX, cours CLIO C4*, Syséca 1987.

[CLIOc] CLIO, Manuel Administrateur : *Sécurité Intégrité*, CF4.2-VAX-A5, Syséca Juin 1989.

[CUS89] *CLIO/VAX : Processus de surveillance , Etude Détaillée*, CUSSEY JL. Syséca, Septembre 1989.

[RDBa] VAX RDB/VMS : *Guide to database maintenance and performance*, July 1988.

[RDBb] VAX RDB/VMS : *Introduction and master index*, July 1988.

[RDBc] VAX RDB/VMS : *Reference manual*, July 1988.

[VMSa] Digital,VAX,VMS : *Synchronisation et communication entre Process*, Novembre 1989.

[VMSb] Digital,VAX,VMS : *System Programming*, avril 1988.

[VMSc] Digital,VAX,VMS : *System service, System routines*, avril 1988.

[VMSd] Digital,VAX,Pascal : *User's guide*, mars 1985.

[VMSe] Digital,VAX,Pascal : *Programming in Vax Pascal*, mars 1985.

Jean-luc CUSSEY

Eléments de sécurité du système de gestion de bases de données CLIO/VAX dans un environnement réseau Conception et réalisation du processus de surveillance CLIOPS

Mémoire d'ingénieur C.N.A.M Grenoble 1990

Suite à l'accroissement considérable des configurations informatiques en réseau, est survenu un problème nouveau de sécurité d'exploitation, lié aux défauts de fonctionnement propres au réseau lui-même.

C'est pourquoi il nous est apparu indispensable, dans cet environnement qui se développe, de faire évoluer les mécanismes de sécurité du SGBD (Système de Gestion de Bases de Données) CLIO afin qu'ils puissent contrôler et gérer ce nouveau type d'anomalie.

Notre objectif a donc consisté à pourvoir le Système de Gestion de Bases de Données CLIO (variante DEC-VAX) d'une sécurité de fonctionnement optimale dans une utilisation en environnement réseau.

Cette réflexion et cette étude ont abouti à la conception et à la réalisation d'un processus de surveillance du SGBD CLIO/VAX dénommé CLIOPS.

mots-clés : *Base de Données, CLIO, Merise, processus de surveillance, réseaux, SGBD, sécurité des SGBD, VAX, VMS*

keywords : *Database, CLIO, Merise, survey process, network, DBMS, DBMS security, VAX, VMS*