

REPUBLIQUE DU SENEGAL

---

UNIVERSITE CHEIKH ANTA DIOP - DAKAR

---

**Ecole Nationale Supérieure Universitaire de Technologie**

Année Universitaire 1992 - 1993



**DEPARTEMENT INFORMATIQUE**

**Cycle : Ingénieur**

**MEMOIRE DE FIN D'ÉTUDES**

---

Présenté et soutenu publiquement le 6 Juillet 1993

par : Mademoiselle **Yawa Edem FIANYO**

pour l'obtention du  
**DIPLOME D'INGENIEUR EN INFORMATIQUE**

Sujet :

**ETUDE ET REALISATION D'UN INFO-SERVEUR SUR LA BASE HORIZON**

**Jury d'examen** : Mrs. CHEVILLOTTE Hervé  
CLERCIN Christian  
CORENTHIN Alex  
HENIN Dominique  
MASSON Bertrand  
SECK Tidiane

à Ruth,  
à Sedi,  
à Eyam, Emefa, Eyom

# REMERCIEMENTS

A l'ensemble du département informatique de l'ENSUT :

- à M. A. CORENTHIN, chef du département pour sa disponibilité;
- à tous mes professeurs pour ce qu'ils m'ont enseigné et plus particulièrement à M. C. CLERCIN pour sa disponibilité;
- à mes camarades de promotion pour la bonne ambiance qui a régné pendant ces 3 années et plus particulièrement à Aminata GAYE, pour son amitié;

A mes encadreurs durant ce stage :

- M. H. CHEVILLOTTE, pour la liberté d'action qu'il m'a laissée dans mon travail et pour sa disponibilité;
- Mme V. LEBLANC, responsable du CRDO pour son accueil, sa disponibilité, sa patience;

A tous ceux qui à l'ORSTOM ont contribué à l'aboutissement de ce travail :

- M. P. BOIVIN, responsable du laboratoire de Pédologie,
- M. K. TEPE, stagiaire au laboratoire de Pédologie;

**"Chaque ligne de programme offre la satisfaction de faire des mathématiques pures et en plus de réaliser quelque chose qui fonctionne"**  
**Bill GATES**

Introduction .....	1
1. Le Cahier des Charges .....	2
1.1. Etude de l'existant .....	2
1.1.1. Présentation de l'ORSTOM .....	2
1.1.2. Le R.I.O : Réseau Intertropical d'Ordinateurs .....	5
1.1.3. La documentation à l'ORSTOM.....	18
1.2. Intérêt du nouvel outil par rapport à l'existant .....	26
1.2.1. Motivation, but et services attendus .....	26
1.2.2. Utilisateurs concernés .....	27
1.2.3. Portée et limite .....	27
1.2.4. Suite prévue.....	27
1.3. Les exigences fonctionnelles .....	27
1.3.1. Fonctions de base .....	27
1.3.2. Performances souhaitées .....	28
1.4. Les contraintes d'utilisation .....	28
1.5. Les critères d'appréciation.....	28
1.6. Contraintes imposées et faisabilité technologique.....	29
1.6.1. Moyens .....	29
1.6.2. Configuration matérielle et logicielle .....	29
1.6.3. Environnement de développement .....	30
1.6.4. Configuration de tests .....	30
1.7. L'Etude de faisabilité.....	32
1.7.1. La reconnaissance de l'expéditeur du message .....	32
1.7.2. Le langage LOGOTEL .....	32
1.7.3. Le contrôle des requêtes .....	33
1.8. Les caractéristiques de la solution retenue .....	33
1.8.1. Le contrôle des requêtes .....	34
1.8.2. Le moment de prise en compte de la demande de l'utilisateur .....	35
1.8.3. Le choix du site où sont contrôlées les requêtes .....	36
1.9. Le planning prévisionnel.....	37
2. Le cahier des spécifications .....	39
2.1. Le contexte du développement.....	39
2.1.1. Le mail .....	40
2.1.2. Le sendmail .....	40
2.1.3. LEX.....	41
2.1.4. YACC .....	41
2.3 Les spécifications fonctionnelles.....	43
2.3.1. examen de la boîte aux lettres horizon.....	43

2.3.2. contrôle de la structure du fichier .....	43
2.3.3. contrôle syntaxique des requêtes .....	44
2.3.4. soumission des requêtes correctes .....	44
2.3.5. envoi des réponses à l'utilisateur .....	44
2.4 Les spécifications d'interface : l'interface C/TEXTO .....	45
2.5. Les spécifications opérationnelles .....	46
2.5.1. Les facteurs humains .....	46
2.5.2. Les capacités .....	46
2.5.3. La qualité .....	46
2.5.4. La base de données utilisée .....	48
2.5.5. Les standards : présentation des notices .....	54
3. La conception .....	56
3.1. Liste des modules .....	56
3.2. Conception des modules .....	56
3.2.1. exam_BAL .....	56
3.2.2. infoserveur_horizon .....	58
3.2.3. verif_syntaxe .....	60
3.2.4. traitement-requete .....	63
3.2.5. loginfos2 .....	65
4. L'interface .....	69
4.1. Présentation de l'interface : modules et logiciels mis en jeux .....	69
4.2. Description de la structure de l'interface .....	69
5. Les tests .....	70
5.1. Objectifs .....	70
5.2. La vérification du produit .....	70
5.2.1. exam-BAL .....	71
5.2.2. infoserveur-horizon .....	72
5.2.3. verif-syntaxe .....	72
5.2.4. traitement-requete .....	76
5.2.5. loginfos2 .....	77
5.3. La validation du produit .....	78
6. Le manuel d'utilisation .....	79
6.1. Caractéristiques essentielles de l'info-serveur sur la base HORIZON .....	79
6.2. Comment fonctionne l'info-serveur sur la base HORIZON .....	79
6.3. Structure d'un fichier de requête .....	80
6.4. Comment interroger la base documentaire ORSTOM de son choix .....	80
6.5. Majuscules et minuscules .....	81
6.6. Le contenu du fichier de requêtes .....	81
6.7. Les options possibles .....	81

6.7.1. L'option "aide fichier" .....	81
6.7.2. L'option aide question .....	81
6.7.3. L'option compresse.....	81
6.7.4. L'option format limite .....	82
6.8. Les requêtes.....	82
6.8.1. Les champs interrogeables.....	82
6.8.2. La syntaxe des questions à respecter pour formuler une question .....	82
6.9. Les fichiers réponses de l'info-serveur .....	91
6.9.1. Le messages de réponses obtenues par interrogation de la base documentaire .....	91
6.9.2. Le message de retour des questions syntaxiquement incorrectes .....	91
6.9.3. Le message d'avertissement en cas de compression du fichier de réponses .....	91
6.9.4. Le message d'avertissement en cas de problème survenu lors de l'exécution de l'application. ....	92
6.10 Les messages d'erreurs .....	92
6.10.1. Les messages d'erreur concernant le fichiers de requêtes. ....	92
6.10.2. Les messages concernant les questions posées .....	92
7. Le manuel d'exploitation .....	94
7.1. Configuration informatique nécessaire.....	94
7.2. Messages : signification et remèdes .....	95
7.3. Le manuel d'installation : mise en oeuvre .....	95
7.3.1. Les fichiers sources de l'info-serveur sur la base HORIZON .....	95
7.3.2. Installation de l'info-serveur sur la base HORIZON .....	96
Conclusion .....	100
BIBLIOGRAPHIE .....	101
ANNEXES .....	102

## Introduction

Ce mémoire est le produit d'un travail réalisé lors du stage de fin d'étude de six mois (1er février-1er août 1993) nécessaire à l'obtention du diplôme d'Ingénieur en Informatique de l'ENSUT.

Ce stage s'est déroulé à L'Unité Locale d'Informatique Scientifique (ULIS) de l'ORSTOM à Dakar, en collaboration avec le Centre Régional de Documentation ORSTOM (CRDO). Il avait pour objet l'étude et la réalisation d'un info-serveur sur la base documentaire de l'ORSTOM.

L'info-serveur est un système de base de données dont la consultation s'effectue en temps différé. L'utilisateur envoie un message à l'adresse de l'info-serveur. Celui-ci interprète ce message et renvoie à l'utilisateur le résultat de sa requête, via la messagerie électronique.

L'ORSTOM est un institut français de recherche implanté dans plusieurs pays. Il possède un Réseau Intertropical d'Ordinateur (le RIO). La base documentaire de l'ORSTOM est une base de données informatisée qui contient toutes les références bibliographiques des ouvrages publiés par l'ORSTOM depuis sa création. Cette base est connectée au RIO.

Actuellement, le contenu de cette base peut être consulté en mode interactif par les utilisateurs du Réseau Intertropical d'Ordinateur. Cependant cette consultation n'est possible que pour certains utilisateurs du RIO : ceux qui utilisent un protocole de communication leur permettant d'accéder en mode terminal au site de la base de données.

La problématique du stage a été de créer une application qui permette à tous les utilisateurs de consulter la base documentaire, quel que soit leur mode de connexion au RIO. Une telle application présente pour l'utilisateur l'intérêt d'offrir un service moins cher que la consultation interactive.

Ce mémoire présente en détail les phases du cycle de développement du produit réalisé.

# 1. Le Cahier des Charges

## 1.1. Etude de l'existant

### 1.1.1. Présentation de l'ORSTOM

L'ORSTOM (ex Office de Recherche Scientifique et Technique d'Outre-Mer) est l'Institut français de recherche scientifique pour le développement en coopération. C'est un organisme français de recherche scientifique intervenant sur la zone intertropicale avec des compétences et une pratique pluridisciplinaires qui s'exercent en coopération.

Cet établissement public placé sous la tutelle conjointe des ministères français chargés de la recherche et de la coopération se caractérise en chiffres par :

- un budget de 900 millions de FF,
- 880 chercheurs et ingénieurs,
- 720 techniciens et administratifs,
- 1500 personnes recrutées dans les 30 pays partenaires où est représenté l'Institut,
- 41 unités de recherche :
- 17 implantations en Afrique et dans l'Océan Indien,
- 9 en Amérique latine,
- 5 en Asie et dans l'Océan Pacifique,
- 10 en France métropolitaine et dans les DOM-TOM.

La mission essentielle de l'ORSTOM est de contribuer à la connaissance scientifique et à la mise en valeur des milieux physiques, biologiques et humains de sa zone d'investigation à travers des programmes pluridisciplinaires articulés autour de 4 axes prioritaires :

- \* compréhension du fonctionnement des grands écosystèmes et préservation de l'environnement,
- \* conditions d'une agriculture viable à long terme dans les milieux tropicaux fragiles,
- \* environnement et santé publique,
- \* évolution des milieux, dynamique des sociétés et des économies nationales.

Les chercheurs de l'ORSTOM assurent la formation de jeunes chercheurs français et étrangers, dans le cadre de programmes de recherche conduits sur le terrain et dans ses laboratoires ou ceux de ses partenaires.

Tout chercheur ORSTOM est rattaché à un des cinq départements scientifiques de l'Institut:

- Terre, Océan, Atmosphère ;
- Eaux continentales ;
- Milieux et Activité agricole

- Société, Urbanisation, Développement ;

- Santé;

et peut bénéficier du travail des missions et directions d'appui technique :

- Télédétection, Télétransmission qui s'occupe de réception satellite, de traitement d'image, de surveillance de l'environnement marin et climatique, de l'établissement des cartes de végétation, du suivi de la saison cyclonique, de la cartographie des sols;

- La Mission Technique Informatique chargée principalement de la mise en place, de la gestion et du développement du Réseau Intertropical d'Ordinateurs (R.I.O), mais aussi de la recherche en Intelligence Artificielle, de l'établissement de banque de données.

- L'appui technique maritime:

L'ORSTOM dispose de :

deux navires de façade : l'Alis et l'André Nizery;

deux vedettes équipées pour la recherche dans les lagons (Dawa et Coris);

l'ORSTOM participe aussi à l'armement de navires de recherche scientifique étrangers :

le Louis Sauger (Sénégal)

le N'Diago (Mauritanie);

mais aussi à l'entretien de vedettes lagunaires à Abidjan (Côte d'Ivoire).

- La Direction de l'Information Scientifique (DIST) poursuit une politique active de diffusion et de valorisation des résultats de la recherche par l'intermédiaire de trois secteurs : la Documentation, les Editions et la Culture Scientifique et audiovisuelle.

La DIST est responsable de l'édition des périodiques (au nombre de sept), des ouvrages scientifiques, des thèses, des ouvrages documentaires, des actes de colloques, des atlas de structures agraires, des cartes; et aussi de la production des films, des disques, des disques compacts, des cassettes audiophoniques, ainsi qu' une revue de vulgarisation scientifique "ORSTOM-Actualités".

Le travail de l'Institut est régulièrement évalué par des commissions.

Il y a sept commissions scientifiques :

\* géologie-géophysique ;

\* hydrologie-pédologie ;

\* hydrobiologie et océanographie ;

\* sciences du monde végétal et animal;

\* sciences biologiques et biochimiques appliquées à l'homme ;

\* sciences sociales ;

\* sciences de l'ingénieur et de la communication.

Une commission spéciale suit les activités techniques et administratives.

#### 1.1.1.1. L'ORSTOM au Sénégal

L'ORSTOM intervient au Sénégal depuis 1949. Les travaux orientés initialement vers l'inventaire des ressources du milieu naturel sahélien, se sont diversifiés au cours des ans. Actuellement, les 5 Départements scientifiques de l'Institut y sont largement représentés.

Les programmes de recherche conduits en liaison étroite avec les structures sénégalaises abordent les préoccupations nationales : dégradation et régénération du patrimoine naturel, gestion des ressources en eau et en sol, gestion du milieu marin, amélioration des cultures en vue de l'autosuffisance alimentaire, santé, croissance démographique, urbanisation.

Le personnel chargé de réaliser ces programmes de recherche compte 250 agents, chercheurs, ingénieurs et techniciens.

Il se répartit dans trois implantations:

- le Centre ORSTOM de Hann (Sciences de la Terre, Ecologie, Nutrition) ;
- le Centre ORSTOM/ISRA de Bel Air (Biologie des Sols, Nématologie, Sciences Sociales, Santé et Océanographie Physique) ;
- la station de M'Bour (Zoologie et observatoires de Géophysique).

Certaines unités de l'ORSTOM sont accueillies par des instituts de recherche sénégalais :

- le CRODT : Centre de Recherches Océanographiques de Dakar Thiaroye de l'Institut Sénégalais de Recherches Agricoles (Océanographie, Hydrobiologie et Télédétection) ;
- l'institut Pasteur (Entomologie médicale et Virologie) ;
- l'ENSUT (Biotechnologies des Fermentations) ;
- l'URCI (Unité de Recherches Communes en Culture In-vitro) : laboratoire commun ORSTOM/ISRA. Cette structure ouverte aux universitaires, est chargée en priorité de réaliser des recherches sur la production de vitro plants d'espèces ligneuses..

Plusieurs unités techniques renforcent le dispositif de recherche :

- l'ULIS (Unité Locale d'Informatique Scientifique) ;
- l'UTIS (Unité de Traitement des Images Satellitaires) ;
- le laboratoire de chimie ;
- le CRDO (Centre Régional de Documentation ORSTOM).

L'ORSTOM poursuit une politique de formation à la Recherche par la Recherche. A ce titre, plusieurs chercheurs de l'ORSTOM participent aux enseignements de l'Université Cheikh Anta Diop de Dakar, des Ecoles d'Ingénieurs et des Organisations Internationales.

L'info-serveur sur la base documentaire HORIZON de l'ORSTOM est un nouveau service de l'Institut, entrant dans le cadre de l'appui technique aux scientifiques.

La mise en place de l'info-serveur doit répondre aux besoins de recherche documentaire des chercheurs. Il doit faciliter la quête de l'information d'un utilisateur en s'affranchissant des contraintes d'espace.

Cela met en jeu deux notions :

- la notion de réseau,
- la notion de documentation ;

qui sont prises en compte à l'ORSTOM par deux entités :

- le R.I.O ou Réseau Intertropical d'Ordinateurs,
- la Documentation ORSTOM.

Afin de bien situer le cadre de l'application, ces deux entités sont présentées dans les paragraphes suivants, avec une étude des aspects intéressant l'application à construire.

### **1.1.2. Le R.I.O : Réseau Intertropical d'Ordinateurs**

Le Réseau Intertropical d'Ordinateurs est un réseau recherche-éducation qui relie actuellement des centres de recherche dans une quinzaine de pays en développement.

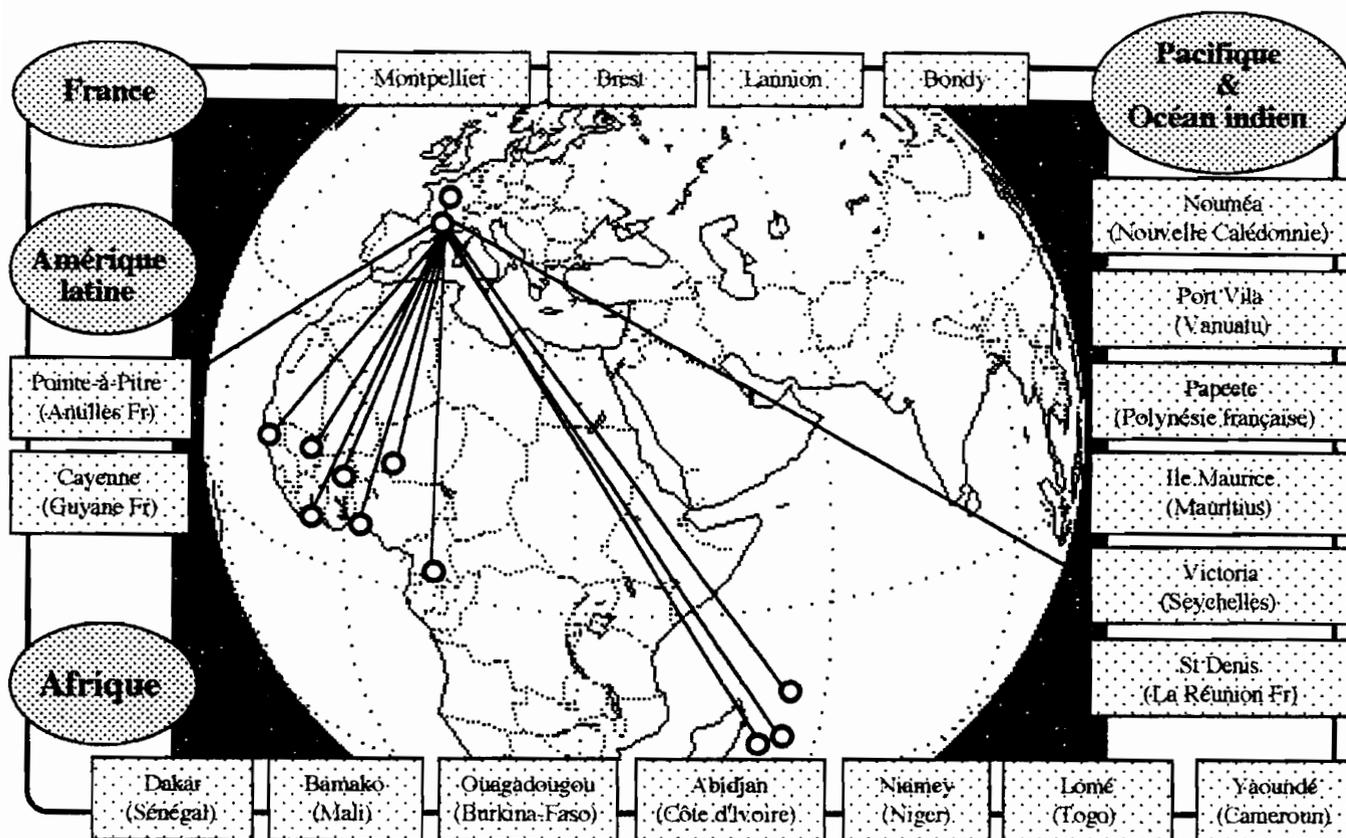
Il est né pour répondre au besoin croissant de communication des structures de recherche des pays en développement, en leur permettant :

- de dialoguer de façon fréquente, non protocolaire, non contraignante avec des institutions internationales et des laboratoires des pays du nord,
- de se conseiller mutuellement dans le montage d'une expérience, de collaborer dans la rédaction d'un article ou d'un rapport, de se transmettre des fiches bibliographiques, de faire exécuter ailleurs des traitements informatiques lourds, dans le cadre d'un projet en commun avec des groupes de recherche dispersés entre plusieurs pays ou continents ;
- de coopérer avec d'autres unités travaillant dans des conditions semblables.

Le RIO est partie prenante de l'Internet (Annexe A) et communique avec tous les grands réseaux internationaux de la recherche et de l'Enseignement Supérieur.

### 1.1.2.1. L'Organisation générale du réseau

Le R.I.O est construit en étoile autour d'un noeud central installé à Montpellier. Dans chaque pays relié, un site distant communique directement avec le site central. Cette organisation procure une bonne fiabilité et optimise les coûts de télécommunication.



### 1.1.2.2. Les protocoles de télécommunication

Les protocoles de télécommunication ont été choisis de manière à :

- fiabiliser le système,
- minimiser le coût de fonctionnement de manière à en autoriser l'accès à tous,
- garantir les délais de distribution. Par exemple, le courrier électronique a un délai de distribution comparable à celui de la télécopie.

Le réseau X25 est le plus souvent utilisé sauf lorsqu'il est difficile ou coûteux d'établir une liaison directe vers un concentrateur. Dans ce cas l'accès RTC (Réseau Téléphonique Commuté)

vers un PAD<sup>1</sup> public est préféré.

Il y a cinq solutions pour faire partie du R.I.O. Elles sont décrites ci-dessous de la plus confortable à la plus légère. Elles s'appuient sur les protocoles de système UNIX : UUCP (Unix to Unix CoPy) et TCP/IP (Transport Control Protocol/Internet Protocol).. (Cf. Annexe B)

#### **1ère solution** : X25 sur ligne directe, protocole IP

matériel : Sun3 ou Sun4 sous SunOS 4.

logiciels commerciaux : SunNet X25<sup>TM</sup>.

Une sortie RS423<sup>2</sup> du Sun est reliée directement au modem synchrone. La gestion du protocole X25 est assurée par le logiciel SunNet X25. La liaison avec les autres sites du R.I.O se fait sous IP.

La liaison sous IP nécessite une ligne X25 d'un débit égal ou supérieur à 9600 bps.(bits par sec.) En revanche c'est une solution qui apporte le maximum de confort. Les ordinateurs reliés de cette manière se comportent comme s'ils étaient sur un même réseau local TCP/IP (au débit près). Il devient possible d'utiliser les commandes des réseaux TCP/IP : *telnet*<sup>3</sup>,

---

<sup>1</sup>Packed Assembling Disassembling : point de jonction entre le RTC et le réseau X25, dont la principale fonction est reconstituer les paquets en message lorsque le transport se fait dans le sens X25-RTC et désassembler les messages en données lorsque le sens de la communication est RTC-X25.

<sup>2</sup>standard de l'EIA (Electric International Agency) reprise par l'ISO (International Standard Organization) pour la norme X26. Caractérise les circuits dissymétriques permettant les bas et moyens débits.

<sup>3</sup>Terminal Network : accès en mode terminal sur une autre Unité Centrale, à travers le réseau

FTP<sup>4</sup>...Les messages sont transmis par *SMTP (Simple Mail Transfert Protocol)* de manière immédiate vers le site destinataire.

Cette solution est relativement onéreuse, elle induit des coûts fixes (l'abonnement X25) et des coûts de fonctionnement plus élevés : nombreux établissements de CV<sup>5</sup>, surplus de caractères générés par "l'encapsulation" des paquets IP dans des paquets X25. Elle est en service sur des lignes spécialisées entre les sites R.I.O de Montpellier, Bondy et Paris, ainsi que dans les liaisons du R.I.O avec l'INRIA (Institut National de Recherche en Informatique et en Automatique) de Roquencourt et le CNUC (Centre National Universitaire) de Montpellier.

**2ème solution** : X25 sur ligne directe, protocole UUCP/F

matériel : Sun3 ou Sun4 sous SunOS 4

logiciels commerciaux : SunNet X25<sup>TM</sup>

Comme dans le cas précédent, une sortie RS423 du Sun est reliée directement au modem synchrone, la gestion du protocole X25 est assurée par le logiciel SunNet X25.

Le protocole UUCP/F est une version d'UUCP destiné aux lignes fiabilisées. Il est bien adapté à la transmission par X25 quelqu'en soit le débit. Comme toute liaison par UUCP, elle est de type "store and forward", les données à transmettre sont stockées sur le site puis envoyées.

L'envoi est effectué soit à chaque requête, soit à une heure déterminée, l'ensemble des requête étant groupées dans un même lot.

Cette solution permet d'obtenir les coûts de transmission les plus bas, le temps d'occupation du circuit virtuel (CV) est minimum alors que UUCP/F génère très peu de caractères supplémentaires (un seul "check sum" par fichier).

Elle est en service à Dakar, Lomé, Niamey, Ouagadougou, Abidjan, Nouméa.

---

<sup>4</sup>File Transfert Protocole :Transfert de fichiers en mode interactif.

<sup>5</sup>Circuit Virtuel : nom donné à une liaison entre 2 équipements de transmission de données dans un réseau X25. L'établissement du CV donne lieu à une facturation forfaitaire.

**3ème solution** : X25 asynchrone

Il s'agit d'atteindre un PAD public à travers une ligne RTC ou une liaison directe de type asynchrone V22.

matériel : Sun3 ou Sun4 sous SunOS 4.0.x

Une sortie RS423 du Sun est reliée au modem asynchrone. La liaison avec les sites du R.I.O se fait sous UUCP protocole F si la ligne vers le PAD est de bonne qualité, protocole G sinon.

Cette solution donne de bons résultats, sans exiger d'investissement puisqu'on se contente d'un point d'entrée sur un PAD public (en général un abonnement peu onéreux).

Cette solution est en service à Cayenne, aux Seychelles, en Guadeloupe et à Tahiti.

**4ème solution** : RTC

matériel : toute machine UNIX

C'est la solution UUCP classique.

Une sortie asynchrone classique de l'ordinateur est reliée au modem asynchrone. La liaison avec les autres sites du R.I.O se fait sous UUCP protocole G.

Elle est en service à Bamako, Bobo-Dioulasso, Port-Villa, Brazzaville, aux Seychelles, et Dakar-ENSUT laboratoire physique-atmosphère.

Le coût de cette solution dépend de la qualité de la ligne. Les connexions échouent lorsque des caractères parasites sont reçus dans la phase d'établissement de la liaison. Il est alors nécessaire de relancer UUCP. Une fois la liaison UUCP établie, la répétition des paquets rallonge le temps des transmission.

Une amélioration très sensible peut être obtenue de 2 manières :

- par l'utilisation de modem à correction et compression (NMP5, V42, V32, V32bis) avec UUCP/F seul capable de gérer le contrôle de flux.
- par l'utilisation de modem de haut débit, intégrant le protocole UUCP (Telebit TrailBlazer Plus™).

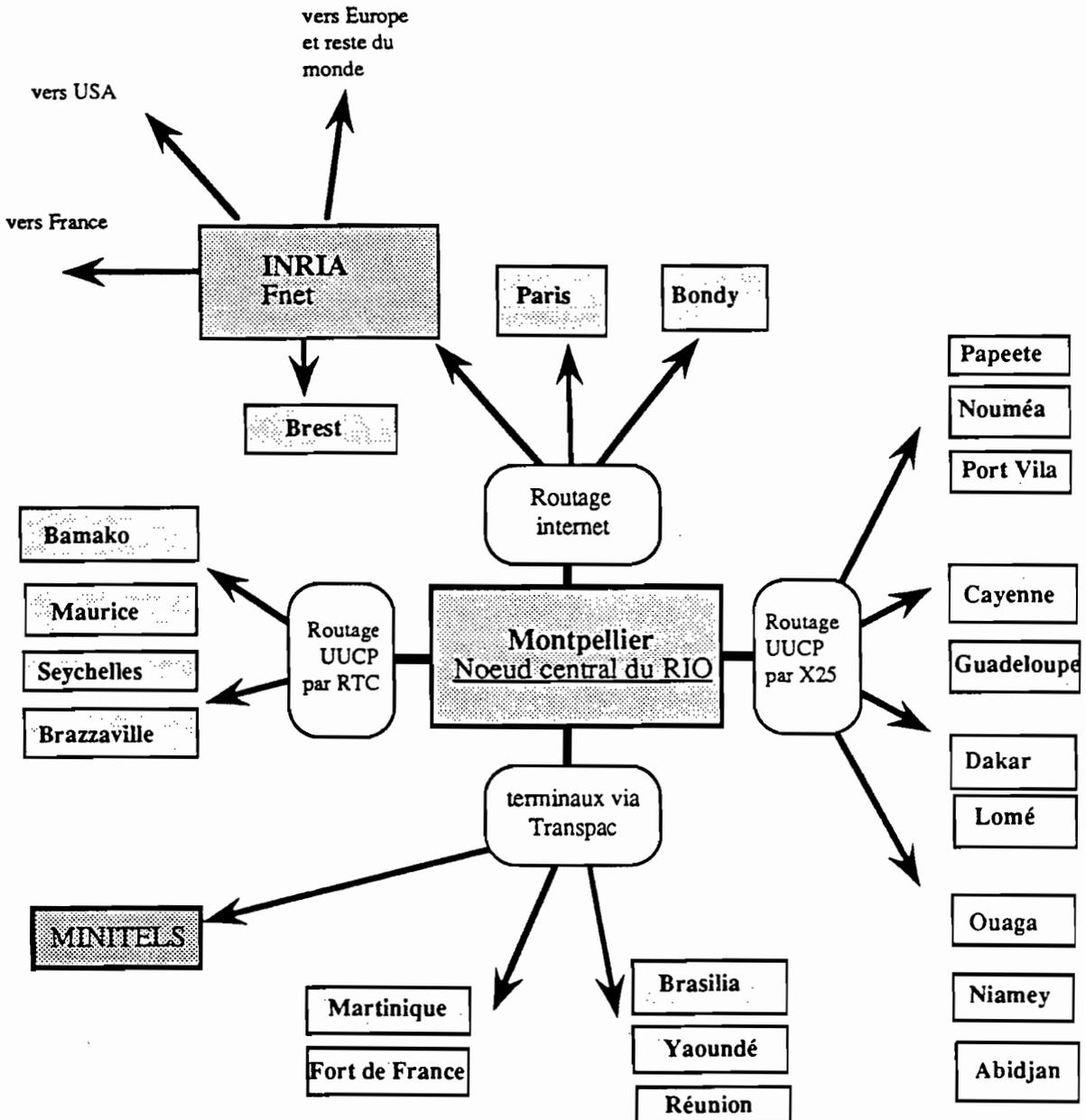
**5ème solution** : UUCP sous DOS (UUPC)

matériel : compatible PC sous DOS

Il s'agit d'une variante intéressante de la solution 4. Elle n'utilise qu'une machine DOS qui émule le protocole UUCP ; celle-ci se comporte vis à vis du site distant comme une

machine UNIX, cependant l'initiative de la liaison appartient nécessairement à la machine DOS.

Cette solution est en service en Equateur, à l'île Maurice, au Mexique, à Pointe-Noire au Congo, et à Dakar-ENSUT département informatique. Elle est utilisée comme backup en cas de panne d'un noeud Unix.



### 1.1.2.3. L'Intégration des réseaux locaux dans le schéma général

La demande d'un réseau local, de la part des utilisateurs de micro-ordinateurs est croissante. Ces réseaux de micros permettent d'améliorer le travail en équipe, d'économiser certains périphériques coûteux (imprimantes laser, traceurs ...), d'exploiter des applications capables de partager des données. Cependant, le coût de l'installation d'un réseau local de micro-ordinateurs est relativement élevé et nécessite un minimum de compétence pour être convenablement administré.

La solution retenue sur le R.I.O consiste à utiliser le système UNIX qui assure la liaison télécom avec les autres sites en tant que serveur de réseau local. Ce réseau dispose alors de fonctionnalités supplémentaires pour un coût marginal moindre :

- \* communication de données (messages, fichiers), de micro à micro, entre les sites ;
- \* accès à partir du micro, aux ressources de la machine UNIX (périphériques, logiciels) ;
- \* sécurité d'accès garantie par le système UNIX.

Les services réseaux sont assurés par les protocoles Ethernet, TCP/IP (fonctions terminal distant et transfert de fichiers) et PCNFS™ (serveurs de fichiers et partage d'imprimante sous DOS).

Le serveur UNIX est "vu" par les Macintosh comme un serveur Apple-Share tandis qu'il est perçu par les PC sous MS-DOS™ comme un serveur NetBIOS™.

### 1.1.2.4. Les services du RIO

#### 1.1.2.4.1. La messagerie électronique

La messagerie est le service de base de tous les réseaux longue distance. Elle permet de transmettre en toute confidentialité, des textes de quelques lignes à plus de 50 pages.

Le texte est saisi sur un poste du réseau ou sur minitel. Il peut être rédigé à l'aide d'un traitement de texte. Le destinataire reçoit le message dans une *boite aux lettres* sur son poste de travail habituel (micro, mini-ordinateur ou minitel). Une fois lu, le courrier peut être conservé dans un fichier informatique et retravaillé avec un traitement de texte.

La messagerie électronique permet de correspondre avec tous les utilisateurs des réseaux internationaux : Internet, Fnet, EUnet, EARN, Bitnet, UUnet... Au total elle relie plusieurs millions de correspondants dans les secteurs de la recherche, de l'enseignement supérieur et des industries de pointe.

Chaque utilisateur dispose d'une boîte aux lettres. Celle-ci est repérée par une *adresse électronique* unique sur l'ensemble des réseaux. L'expéditeur d'un message est identifié par un mot de passe individuel qui garantit l'authenticité et la confidentialité du courrier. La messagerie inclut le service de publipostage (distribution à plusieurs correspondants) et de réexpédition automatique.

### **Les accents**

La messagerie du R.I.O est aux normes RFC822. Celles-ci n'incluent à priori que les caractères purement ASCII (7bits). Pour les utilisateurs de langue française, l'absence d'accent est ressentie comme une contrainte pénible et nuit à l'intelligibilité du texte. C'est pourquoi une solution d'implémentation des caractères accentués français a été cherchée. Elle est réalisée par des interfaces utilisateurs développés à l'ORSTOM<sup>6</sup> : PCMESSOR, NFSMESSOR, UUMESSOR et MESSORTELE.

Les caractères accentués restent codés sur 7 bits, ce qui garantit la compatibilité avec les autres réseaux mais l'interprétation de certains caractères ({,},...) est modifiée conformément à la norme NRC. Cette implémentation est compatible avec le minitel bi-standard en mode "téléinformatique, terminal français".

### **L'annuaire ORSTOM**

L'annuaire du RIO contient les adresses électroniques de toutes les personnes de l'ORSTOM ayant une boîte aux lettres.

#### **1.1.2.4.2. La transmission de documents**

Le service messagerie se limite à l'envoi de texte. Par transmission de documents, il faut entendre l'envoi de rapports "balisés", c'est à dire préparés avec un logiciel de traitement de texte, contenant diverses polices et corps de caractères, des indications de styles, de justification...

Le principe est le suivant : le document est codé dans un format intermédiaire et intégré dans un message, il sera reconverti à l'arrivée sous sa forme primitive. Cette technique permet de transmettre des rapports d'environ 50 pages, entièrement balisés enrichis de graphiques et de dessins réalisés avec un logiciel de traitement de texte standard (sur

---

<sup>6</sup>MESSOR : MESSagerie ORstom

Macintosh ou PC compatible IBM). Les documents peuvent être repris et modifiés par le destinataire disposant du même logiciel ou d'un produit compatible avec lui.

#### 1.1.2.4.3. Les forums

Le forum électronique est un outil de débat scientifique. C'est l'équivalent en terme de messagerie de la "communication à plusieurs" sur le téléphone. Les contributions au forum se font par l'envoi de messages qui sont automatiquement distribués à tous les participants.

Le forum est constitué par un groupe d'utilisateurs du R.I.O ayant chacun une boîte aux lettres. Leurs adresses sont enregistrées dans une liste de distribution. Le forum porte le nom de cette liste.

Chaque forum est placé sous la responsabilité d'un animateur. C'est lui qui en demande l'ouverture et qui est à même de répondre à toutes les questions concernant le sujet, les participants et la nature des contributions attendues. Les forums actifs sur le R.I.O sont les suivants :

<b>Nom du forum</b>	<b>Objet de la discussion</b>
agrégation	comportement agrégatif des poisson pélagiques
cyel	cycle de l'énergie dans les lagons d'atolls de Polynésie
ecosys	écosystèmes coralliens
geofish	utilisation de géostatistique en biologie marine
malaria	étude de la malaria
nacre	biologie et environnement de l'huître nacrrière
popsant	relation entre les problèmes de santé et la démographie en Afrique
sas	logiciel de statistiques
sti	sciences et techniques de l'informatique.

#### 1.1.2.4.4. Les info-serveurs

Les info-serveurs peuvent être "physiquement" installés sur tout noeud du R.I.O et sont consultables à partir de tout accès aux réseaux "recherche-éducation", soit plusieurs millions d'accès dans le monde. L'accès n'étant pas réalisé en temps réel, le coût de consultation est réduit au minimum. (environ 200 CFA la page reçue).

Les info-serveurs suivant sont en service sur le R.I.O :

- à chaque forum est attaché un info-serveur qui permet de retrouver un texte soumis au forum grâce à son numéro, ou de consulter la liste des contributions. (tous les textes du forum y sont référencés).

- CNUED: base de données des textes officiels (traduits en français) de contribution à la Conférence des Nations Unies sur l'Environnement et le Développement de Rio de Janeiro en juin 1992 (sommet de la Terre). Ces textes récapitulés dans une liste sont référencés par des numéros.

#### 1.1.2.5. Les utilisateurs

Le nombre total d'utilisateur dépasse 800, l'essentiel est formé par les chercheurs de l'ORSTOM et d'organismes partenaires.

Le trafic est actuellement d'environ 7 Mo (7 Millions de caractères) par mois dont la moitié vers l'Afrique.

##### 1.1.2.5.1. Répartition par spécialité

Discipline scientifique ou activité	nombre de comptes
Sciences humaines	74
Agronomie	72
Sciences de la terre	110
Santé	46
Hydrobiologie / océanographie	145
Statistiques, biométrie, informatique et Télédétection	90
Edition, documentation	35
Administration et directions	150
Non ventilés	80
TOTAL	802

#### 1.1.2.6. La tarification

Un système de tarification interne a été mis en place en 1993.

Est considéré comme consommation de l'utilisateur, le transport, dans la zone RIO des messages qu'il a envoyés et de ceux reçus en provenance d'un autre réseau.

Les utilisateurs financent les dépenses afférentes à leur consommation sur les ressources qui leurs sont alloués (budget d'unité de recherche).

### 1.1.2.7. Le RIO au Sénégal

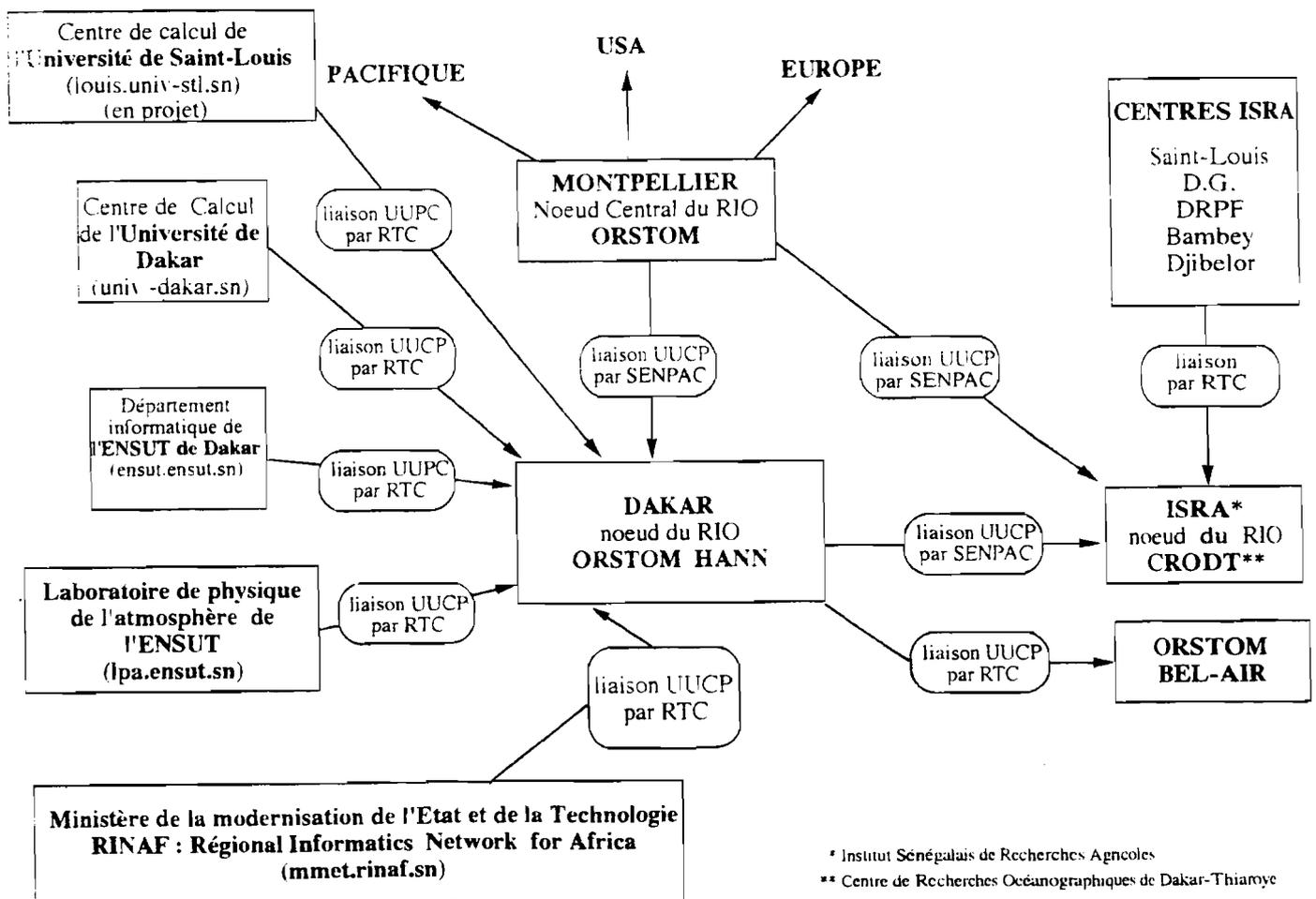
Le Réseau Intertropical d'Ordinateurs au Sénégal s'intègre bien évidemment dans l'organisation générale du RIO.

Il est constitué d'un site principal à Dakar-Hann qui échange des messages avec Montpellier 5 fois par jour ( 7, 10, 12, 14, 16 h ) via SENPAC selon le protocole UUCP/F ( 2ème solution ).

La machine Dakar est serveur de messagerie ou backbone pour d'autres noeuds uucp au Sénégal avec lesquels elle échange les messages plusieurs fois par jour.

#### *L'Organisation générale du réseau*

Le schéma suivant présente les organismes qui font actuellement partie du RIO au Sénégal. Le sens des flèches indique quelles sont les machines qui sont à l'initiative de la communication.



#### 1.1.2.7.1. Le site de Dakar-Hann

Le site principal de Dakar-Hann, dans lequel s'est effectué le stage, utilise la solution UNIX pour assurer la liaison avec les autres sites du RIO. Il assure en plus les fonctions de serveur de réseau local.

La solution réseau local mise en place est une solution Ethernet. Les principales caractéristiques d'un tel réseau sont :

- une architecture en bus multiple sur câble coaxial ;
- une transmission en bande de base à 10 Mbits/s avec un protocole CSMA/CD.

Le protocole CSMA/CD (Carrier Sense Multiple Access/Collision Détection) est utilisé pour résoudre les problèmes d'accès et de droit d'accès au support. Le principe est simple : on écoute avant de parler et ensuite on détecte le plus rapidement possible si une collision s'est produite.

Dans ce cas, on retransmet après un délai aléatoirement suivi.

#### 1.1.2.7.2. disposition topologique

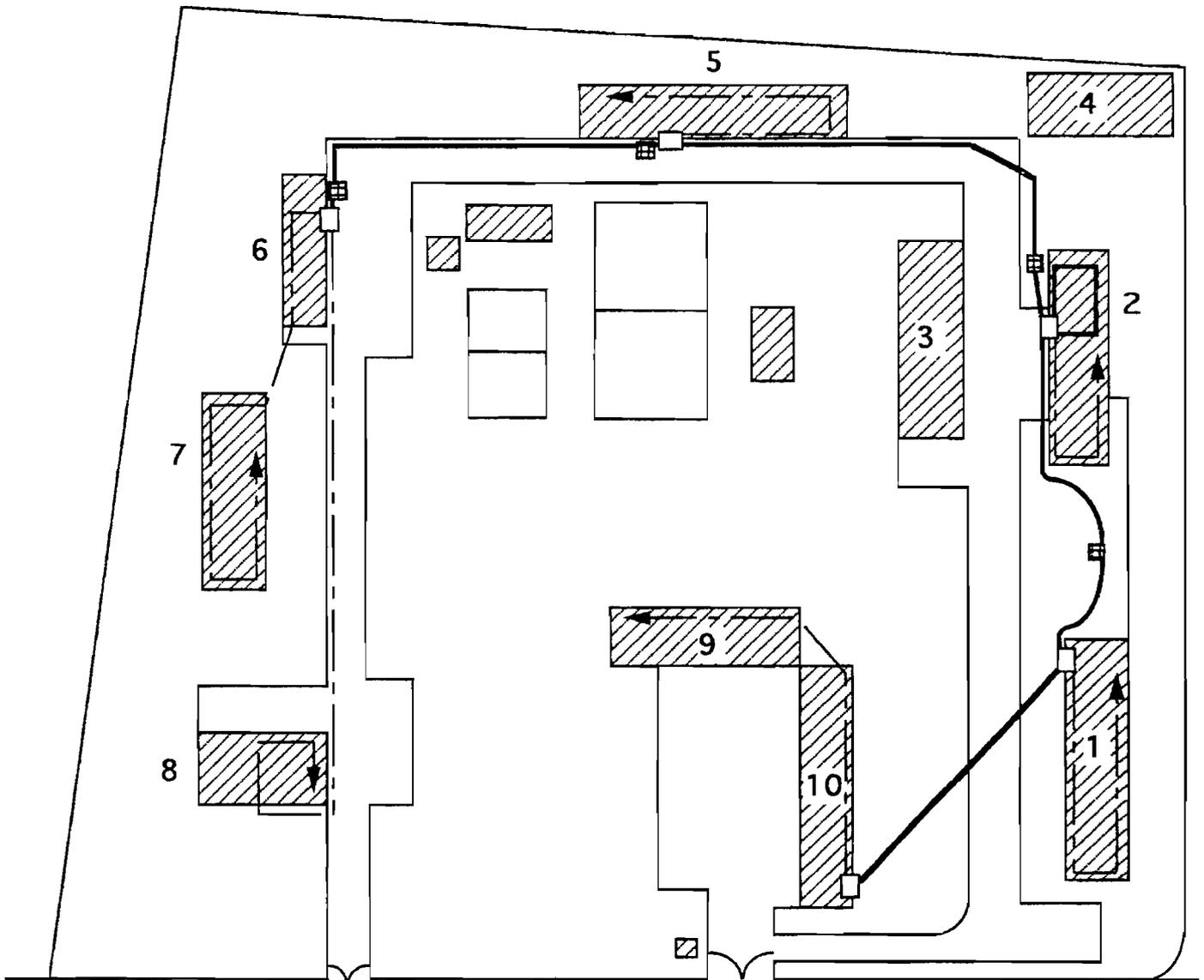
C'est une topologie bus qui associe le Full Ethernet et le Fin Ethernet.

Des segments de câbles coaxiaux relient tous les bâtiments du site.

A l'intérieur d'un bâtiment, le Fin Ethernet rattaché au Full par un répéteur Isolan 1120 permet de connecter au réseau les machines du bâtiment considéré.

Le câble coaxial est très onéreux mais présente l'avantage d'être très fiable.

# RESEAU ETHERNET du centre ORSTOM DAKAR- HANN



ROUTE DES MARISTES

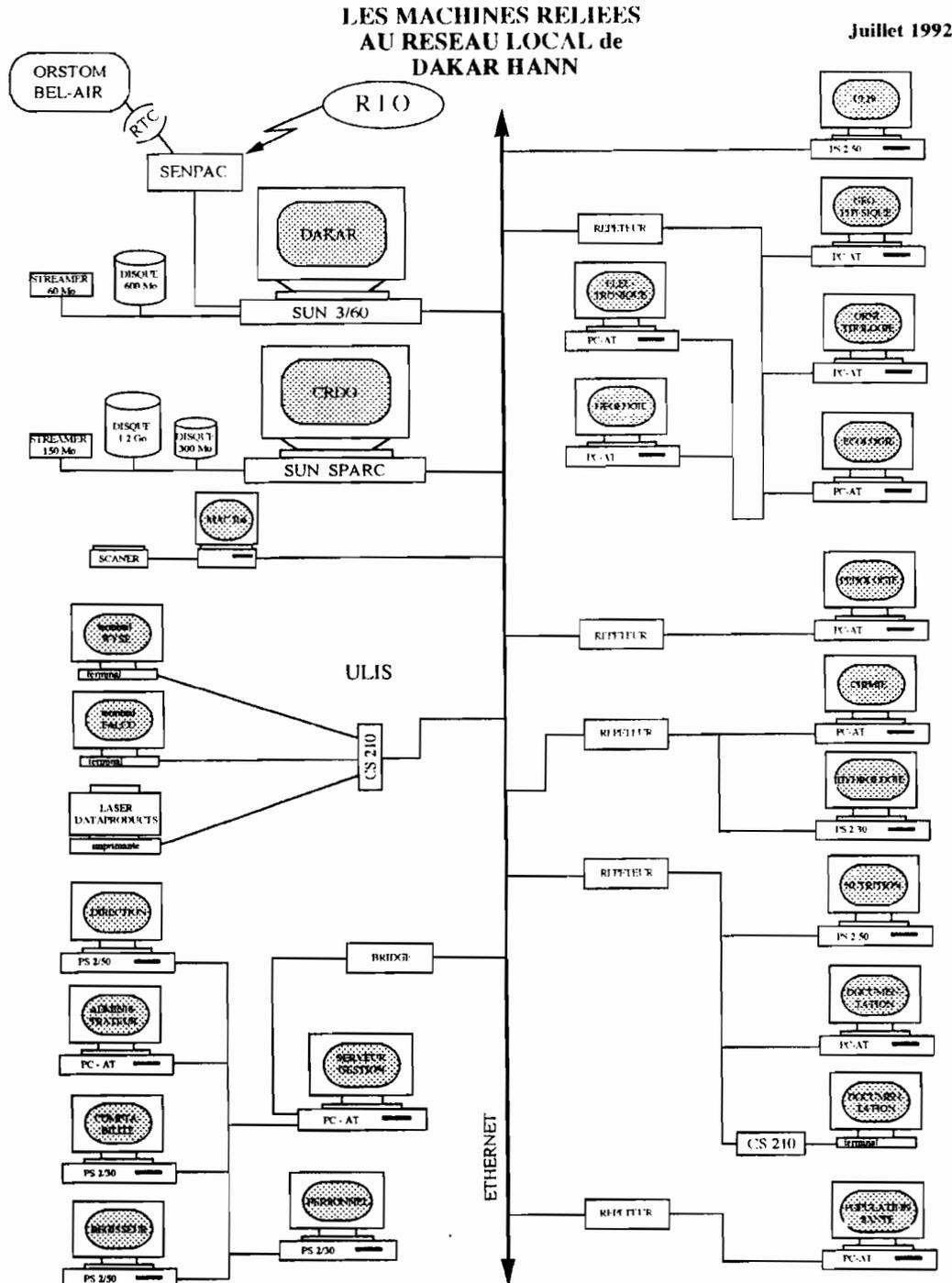
- ETHERNET (câble jaune)
- ▶ ETHERNET  
FIN
- ENTREE DE BATIMENT  
REPETEUR ISOLAN 1120
- ▣ REGARD : JONCTION  
ENTRE 2 SEGMENTS  
ETHERNET JAUNE

- 1 LABORATOIRE DE CHIMIE, HYDROLOGIE
- 2 ULIS, GEOLOGIE, GEOPHY SIQUE, ECOLOGIE,  
ORNITHOLOGIE, ELECTRONIQUE
- 3 PREPARATION ET STOCKAGE
- 4 MAINTENANCE
- 5 PEDOLOGIE
- 6 NUTRITION
- 7 POPULATION ET SANTE
- 8 DOCUMENTATION
- 9 DIRECTION
- 10 ADMINISTRATION

LONGUEUR : 304 m câble jaune  
400 m câble fin

### 1.1.2.7.3. organisation du réseau-local

La solution Ethernet propose comme application les standards TCP/IP et surtout NFS (d'origine Sun). NFS permet en particulier d'accéder de façon transparente et délocalisée à tous les fichiers des ETTD (Equipement Terminal de Traitement de Données).



#### 1.1.2.1.4.7. Les utilisateurs

Répartition par organisme

ORSTOM	46
ENSUT	10
ISRA	13
MMET	3
TOTAL	72

### **1.1.3. La documentation à l'ORSTOM**

#### 1.1.3.1. Les missions

##### 1.1.3.1.1. Appui documentaire aux scientifiques de l'ORSTOM

Les chercheurs doivent être en permanence à l'affût de nouvelles données pouvant susciter de nouvelles idées. Il leur faut sans cesse savoir et comprendre ce qui se passe dans leur domaine scientifique.

Le rôle de la documentation ORSTOM est d'aider les chercheurs à trouver de façon circonstancielle, immédiate, des informations aussi exhaustives et pertinentes que possible. Par ailleurs, elle assure la gestion des abonnements aux revues scientifiques et celle des acquisitions d'ouvrages d'intérêt scientifique.

##### 1.1.3.1.2. Valorisation des travaux de l'ORSTOM

C'est la documentation qui est chargée à l'ORSTOM de rendre accessible au public la production scientifique de l'Institut.

#### 1.1.3.2. La structure

Pour remplir cette mission, la documentation ORSTOM est rattachée à la DIST (Direction de l'Information Scientifique et Technique). Elle est structurée en Centres de Documentation (CD) de deux sortes.

### 1.1.3.2.1. En France :

En France métropolitaine, il existe 3 CD spécialisés :

- Bondy (région parisienne).

Spécialisé dans le domaine des Sciences de la terre, Télédétection et Informatique. C'est aussi l'endroit où est physiquement placé le fonds documentaire HORIZON.

- le CEDID (Paris) : le Centre de Documentation et d'Information scientifique pour le Développement. C'est un CD spécialisé dans le domaine des Sciences Economiques et Sociales.

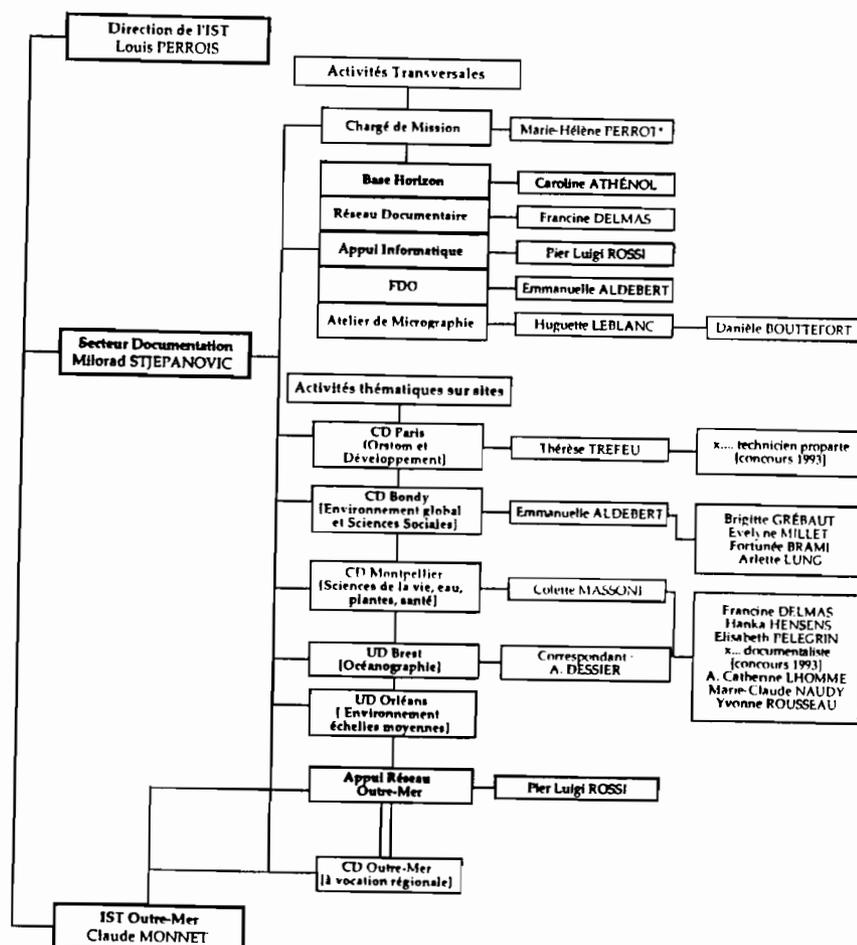
- Montpellier

Spécialisé dans les Sciences du monde végétal et animal, la santé et les ressources halieutiques.

### 1.1.3.2.2. A l'extérieur :

Dans chaque représentation ORSTOM à l'extérieur, se trouve un CD qui s'occupe de la documentation au niveau national.

ORGANIGRAMME DU SECTEUR 1993



\* Vice-présidente de la CS7, Mme FERROT assure également un appui au CEDID et au pôle documentaire sciences sociales.

### **1.1.3.3. Le principal produit de la documentation : le fonds documentaire HORIZON**

Le principal produit de la documentation ORSTOM est la base HORIZON.

C'est une base bibliographique constituée actuellement d'environ 35.000 références dont 90% sont des publications ORSTOM concernant les pays où intervient l'Institut.

De cette base est extrait semestriellement un bulletin bibliographique à caractère multidisciplinaire présentant les dernières parutions.

La base de données documentaire est physiquement implantée à Bondy (région parisienne) et est gérée par le logiciel TEXTO.

HORIZON couvre les domaines d'intérêt de l'Institut :

- Sciences de la Terre,
- Sciences du Monde Végétal et Animal,
- Océanographie et Hydrobiologie,
- Santé,
- Sciences Sociales,
- Sciences de l'Ingénieur et de la Communication.

L'alimentation de la base HORIZON est effectuée par les Centres de Documentation de l'Institut en deux temps.

Dans le souci d'éviter un double travail de traitement documentaire, les Centres de Documentation qui sont rattachés aux représentations ORSTOM à l'extérieur de la France indexent la production locale de leur centre.

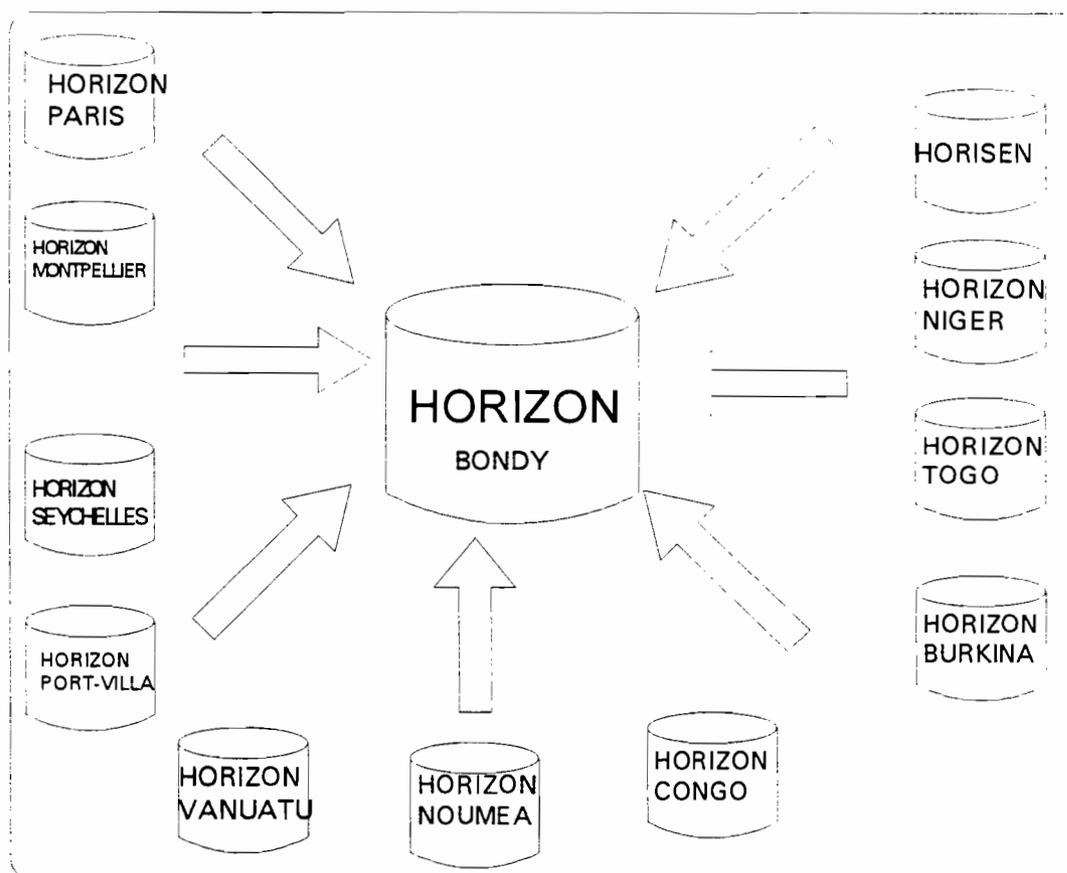
Une fois par mois, ils envoient leurs fichiers à Bondy pour vérification et reversement dans la base HORIZON. Cette vérification de l'indexation est effectuée par les Centres de Documentation spécialisés, qui se chargent d'enregistrer les ouvrages édités par l'ORSTOM.

En plus de la base HORIZON, chaque centre de documentation possède un fonds propre qui contient les références bibliographiques de tous les ouvrages qui y sont physiquement présents, à savoir:

- les publications scientifiques du centre ORSTOM du pays qui seront reversés dans HORIZON,

- les ouvrages acquis par le centre concernant le pays dans lequel est basée la représentation ORSTOM,
- les revues scientifiques reçues par le centre.

Les bases locales ont les mêmes caractéristiques que la base HORIZON, (même structure de fichiers, même logiciel d'exploitation), et ne diffèrent de celle-ci que par le contenu.



#### 1.1.3.4. L'activité des Centres de Documentation : le CRDO de Dakar

Le CRDO ou Centre Régional de Documentation ORSTOM de Dakar est le Centre de Documentation rattaché à la représentation ORSTOM du Sénégal.

C'est un centre de documentation scientifique à vocation régionale dont le domaine d'intérêt couvre l'ensemble des pays sahéliers.

Il est composé de 3 personnes :

Mme V. Leblanc : Responsable du CRDO

M. M. N'Dong : Adjoint au responsable, documentaliste

M. F. Seck : Aide-documentaliste

Il a plusieurs fonctions :

- la gestion des abonnements aux revues scientifiques souscrits par les laboratoires,
- l'acquisition des ouvrages demandés par les laboratoires ou jugés intéressants par le responsable du CRDO,
- l'accueil du public voulant consulter et exploiter sur place les ouvrages du CRDO,
- la gestion du fonds audiovisuel (vidéo) des films produits par l'ORSTOM
- l'alimentation du fonds documentaire HORIZON et la gestion du fonds documentaire propre au CRDO (base HORISEN).

#### 1.1.3.4.1. Les activités du CRDO relative au fonds documentaire

##### *1.1.3.4.1.1. Indexation du contenu des documents*

Tout nouvel ouvrage (publication ORSTOM, achat du centre de documentation) reçoit le traitement documentaire suivant : collationnement, inventaire, catalogage, indexation, saisie de la référence.

Le collationnement consiste à contrôler le document afin de d'assurer qu'il est en parfait état.

L'inventaire consiste à inscrire dans un registre dans l'ordre chronologique de son acquisition, le document destiné à être conservé dans le fonds documentaire.

Le catalogage et l'indexation se font à travers le renseignement d'un bordereau documentaire d'analyse-indexation.

Ce bordereau (voir annexe) contient le nom de tous les champs de la base documentaire à remplir.

Le documentaliste indique sur le bordereau les caractères signalétiques de la publication (mention de l'auteur, titre, éditeur, nombre de pages, etc.). C'est ce que l'on appelle le catalogage.

Dans un deuxième temps, à l'aide du vocabulaire (liste de termes thématiques classés par ordre alphabétique), le documentaliste, à travers des termes sélectionnés appelés mots-clés doit refléter le contenu de l'ouvrage. C'est ce que l'on appelle l'indexation.

L'indexation par mots-clés a pour but de décrire le sujet d'un document à partir de son contenu. Cette analyse se fait à l'aide de mots documentaires : les descripteurs ou candidats descripteurs.

L'utilisation des mots-clés pour l'analyse d'un contenu informatif permet :

- une grande finesse de description,
- une facilité de description pour un document recouvrant plusieurs domaines d'intérêt différents,
- une facilité d'établissement des liens de synonymie ou de hiérarchie. (pour le thesaurus).

L'indexation des documents est un travail d'une importance primordiale parce qu'il conditionne l'efficacité des résultats d'interrogation. La pertinence des réponses en dépend, quelque soit le degré de performance du système d'interrogation.

Les règles d'indexation suivies au CRDO, sont les même dans tous les centres ORSTOM. Les plus générales sont :

- utilisation du vocabulaire multidisciplinaire HORIZON<sup>7</sup> pour l'indexation des mots-clés;
- utilisation des lexiques et des thesaurus<sup>8</sup> IBISCUS<sup>9</sup> (lexique régional, des organismes associés au développement) pour l'indexation des organismes, des localités traitées par le document;
- utilisation du lexique ethnique et du lexique langue du CNRS (francis) pour l'indexation des langues traitées par le documents;
- application des règles de normalisation de l'Association Française de NORmalisation (AFNOR) concernant la documentation et notamment les normes:
  - NF Z 40-003 relative à l'acquisition et au traitement des documents;
  - NF Z 41-001 pour la présentation des publications périodiques;
  - NF Z 41-004 en concordance technique avec la norme ISO 2384 relative à la présentation des documents traduits;
  - NF Z 41-033 pour la présentation des articles de périodiques;

---

<sup>7</sup>vocabulaire HORIZON liste alphabétique et multidisciplinaire de mots-clés reflétant les programmes menés par l'Institut. Chaque terme a une définition simple et précise reconnue avec la même notion par tout lecteur.

<sup>8</sup>Un thesaurus est une liste structurée et hiérarchisée de mots-clés.

<sup>9</sup>IBISCUS base de données spécialisée dans le domaine des sciences économiques et sociales appliquées aux problèmes de développement. Elle est alimentée par des organismes s'intéressant au Tiers-monde : Bureau pour le Développement de la Production Agricole (BDPA), Centre International de Recherche Agronomique pour le Développement (CIRAD), ORSTOM. Créée sur l'initiative du ministère français de coopération.

- NF Z 47-102 relatives aux principes généraux pour l'indexation des documents (voir Annexe).

#### *1.1.3.4.1.2. Gestion de la base documentaire*

Tout nouveau bordereau dûment rempli par le documentaliste est saisi dans un fichier temporaire. Ce fichier est relu systématiquement. Des corrections y sont faites : fautes de frappe, harmonisation de l'indexation...

Après validation des références, le fichier temporaire est reversé dans le fichier principal de la base de données locale : HORISEN.

Le logiciel d'exploitation de la base documentaire permet une mise à jour propre du fichier principal avec mise à jour automatique des index associés.

A la demande de Bondy où se trouve l'équipe chargée de l'alimentation de la base HORIZON, les références des nouveaux ouvrages ORSTOM produits par les différents laboratoires du centre ORSTOM de Dakar sont extraits de la base HORISEN et envoyées via le RIO aux centres de documentation spécialisés pour vérification avant reversement dans HORIZON.

Les éventuelles modifications apportées par les centres de documentation spécialisés sont notifiées sur bordereau au CRDO qui se charge de les enregistrer dans la base HORISEN.

#### *1.1.3.4.1.3. Exploitation de la base documentaire*

L'exploitation de la base documentaire prend deux formes :

##### **- la recherche rétrospective**

Il s'agit d'une recherche interactive immédiate sur la base locale (HORISEN) ou distante (HORIZON) effectuée à la demande du public ou des chercheurs ;

##### **- la recherche différée ou diffusion sélective**

Elle permet l'établissement des produits documentaires suivant :

### *- le catalogue*

Le catalogue est l'ensemble des notices catalographiques des documents du fonds documentaire. Il est mis à jour chaque année. Sa finalité est de permettre et de faciliter l'exploitation du fonds documentaire.

Une notice catalographique comprend entre autre la notice bibliographique, et la cote établie conformément aux principes du catalogue considéré. La cote (ensemble de lettres, chiffres, signes) sert à localiser la place d'un document dans un fonds).

La notice bibliographique contient :

- des éléments signalétiques qui aident à localiser le document recherché ;
- des éléments de "contenu" analytique : résumé et mots descripteurs.

### *- le bulletin bibliographique*

Le bulletin bibliographique se distingue du catalogue par le fait qu'il apparaît :

- sous un volume plus réduit ;
- avec une spécialisation plus grande.

Le bulletin bibliographique du CRDO indique les dernières acquisitions ou parutions disponibles dans le centre documentaire, classées par thèmes.

## 1.2. Intérêt du nouvel outil par rapport à l'existant

### 1.2.1. Motivation, but et services attendus

La base HORIZON et toutes les bases locales des centres ORSTOM ont la même structure. Elles sont installées sur des machines connectées au Réseau Intertropical d'Ordinateurs (R.I.O), et pourraient donc être accédées par l'ensemble des utilisateurs du réseau.

Seule l'interrogation en mode interactif est actuellement possible. Elle est utilisée essentiellement par les centres de documentation qui interrogent en mode interactif la base centrale HORIZON.

Ce mode intéressant pour la satisfaction immédiate de la demande reste cependant cher. On utilise le réseau en mode terminal pour l'envoi de la requête, le traitement de celle-ci et la réception des références bibliographiques.

De plus tous les utilisateurs du R.I.O n'ont pas accès à ce service, car tous les points d'accès du R.I.O ne peuvent pas fonctionner en mode terminal.

Le mode terminal n'est réalisable que lorsqu'une liaison a été établie entre les deux sites communiquant sur RTC ou X25, et que le protocole utilisé offre le service d'accès en mode terminal.

Le premier objectif du projet "info-serveur sur la base HORIZON" est de permettre à tous les utilisateurs du R.I.O de consulter n'importe quelle base documentaire ORSTOM en passant par la messagerie électronique du R.I.O (MESSOR).

Le but est donc de permettre aux non-initiés d'avoir accès au contenu de la base documentaire ORSTOM pour consulter ses productions. Cet accès devra être :

- possible en tout point d'accès du R.I.O,
- peu coûteux grâce au traitement en différé et à la réduction au maximum du temps d'occupation du réseau,
- et contrôlé pour garantir l'intégrité de la base consultée.

### **1.2.2. Utilisateurs concernés**

Les utilisateurs concernés sont les chercheurs de l'ORSTOM et les documentalistes de l'Institut, tous les partenaires scientifiques de l'Institut ayant signé une convention leur permettant d'utiliser le R.I.O : chercheurs d'autres instituts, chercheurs et professeurs d'universités, mais aussi les documentalistes d'autres organismes.

### **1.2.3. Portée et limite**

Le produit à réaliser est un produit générique. L'info-serveur sur la base HORIZON est un produit destiné à être implanté sur toutes les bases ORSTOM possédant un fonds documentaire informatisé.

Ainsi pourront être accédées non seulement la base documentaire centrale de l'ORSTOM : HORIZON (horizon@orstom.orstom.fr), mais aussi tous les fonds documentaires informatisés des centres ORSTOM :

- à Dakar (horizon@dakar.orstom.sn),
- à Ouagadougou (horizon@ouaga.orstom.bf),
- à Nouméa (horizon@noumea.orstom.fr),
- etc.

Le produit réalisé sera d'abord implanté sur le site de Dakar, avant d'être successivement implanté sur le noeud central du R.I.O et sur les autres sites supportant une base locale informatisée.

### **1.2.4. Suite prévue**

Une application vidéotex devrait suivre la réalisation de l'info-serveur, pour permettre à tous les utilisateurs de minitel de consulter la base centrale HORIZON en mode interactif.

## **1.3. Les exigences fonctionnelles**

### **1.3.1. Fonctions de base**

L'application doit remplir les fonctions suivantes :

- transporter un ensemble de requêtes d'un utilisateur au site de support de la base documentaire choisie,

- sélectionner et mettre en forme les références pertinentes pour chacune des requêtes,
- réexpédier les références pertinentes à l'utilisateur.

L'application doit être portable pour une implantation sans problème sur les autres sites.

### **1.3.2. Performances souhaitées**

L'application devra effectuer un contrôle des requêtes afin d'éviter de faire traiter par le logiciel d'exploitation de la base documentaire des requêtes syntaxiquement incorrectes.

L'application sera compacte pour faciliter la mise en oeuvre, et rapide pour limiter les durées toujours longues de mise en forme des références bibliographiques.

## **1.4. Les contraintes d'utilisation**

L'utilisation du logiciel doit être possible à partir de la messagerie électronique du R.I.O.

Cela supprime toute possibilité d'interface conviviale pour l'utilisateur.

La convivialité sera prise en charge par des applications spécifiques au système d'exploitation par lequel on accède au R.I.O (DOS, système d'exploitation des Macintosh, UNIX).

## **1.5. Les critères d'appréciation**

Les critères d'appréciation sont les caractéristiques que l'application à construire doit présenter pour obtenir la satisfaction du client.

La définition de tels critères permet de dégager les facteurs de la qualité du logiciel, c'est à dire les propriétés d'un logiciel durable.

-> Fiabilité et robustesse :

L'info-serveur étant un logiciel transactionnel, il envoie et reçoit de grandes quantités d'information.

Il faut éviter les pertes de données, veiller à la confidentialité et à la sécurité des données et des requêtes des utilisateurs. Il ne faut pas que leurs demandes et leurs réponses soient détruites par d'autres utilisateurs.

Il faut aussi veiller à l'intégrité de la base documentaire : le logiciel ne doit permettre que la consultation.

-> Adaptabilité :

Le produit étant destiné à terme à tous les sites où il existe une base documentaire, il est important que les changements à apporter soient minimum. Si ils sont nécessaires, ils doivent être localisés aux mêmes endroits.

-> Maintenabilité :

Le logiciel de gestion de la base documentaire peut évoluer vers un élargissement des possibilités d'interrogation. L'application à construire doit pouvoir intégrer ces changements sans subir de trop lourdes modifications.

## 1.6. Contraintes imposées et faisabilité technologique

### 1.6.1. Moyens

Le logiciel TEXTO/LOGOTEL s'impose pour la réalisation de la partie interrogation de la base documentaire. C'est en effet ce logiciel qui gère l'ensemble des bases documentaires ORSTOM. Le langage de programmation "LOGOTEL" permet d'écrire des applications d'interrogation.

Les outils de programmation du système UNIX peuvent être utilisés. Cela comprend :

- les générateurs d'analyse lexicale et syntaxique LEX et YACC,
- le langage de programmation C et son vérificateur de syntaxe **lint** permettant de contrôler la portabilité des programmes,
- le langage de commande Shell.

Les outils de mise au point des programmes sous UNIX sont aussi disponibles :

- les outils de débbugage **adb** et **dbx**,
- les outils pour l'évaluation des performances **prof**, **tcov** et **time**.

### 1.6.2. Configuration matérielle et logicielle

#### 1.6.2.1. Configuration cible

L'info-serveur à mettre en place sera implanté sur une station de travail SPARC avec comme caractéristiques techniques :

- un processeur SPARC cadencé à 33 MHz,
- une puissance de 16 MIPS (Millions d'Instructions Par Secondes),
- une mémoire de 16 Mo,
- une interface Ethernet qui rattache la station au réseau local Ethernet du centre,
- deux ports série sur lesquels sont connectés une ligne X25 et un modem à 2400 bps.

La configuration comporte en outre :

- un disque de 1,2 Go,
- un écran 19 pouces couleur,
- un lecteur de cassette de 150 Mo

Le système d'exploitation utilisé est UNIX (SunOS 4.1.3), il supporte une Interface Utilisateur Graphique à fenêtres (OpenWindows) offrant un environnement multitâches avec de nombreuses facilités d'utilisation de la machine.

### 1.6.3. Environnement de développement

L'environnement de développement est le même que l'environnement cible, avec en plus les outils de développement d'UNIX sous OpenWindows et en particulier dbxtool permettant une mise au point très ergonomique.

### 1.6.4. Configuration de tests

Toutes les machines disponibles à l'ULIS (Unité Locale d'Informatique Scientifique) associées à leurs logiciels de communication feront partie de la configuration de test. Cela inclut :

- le **SunSPARC** déjà décrit,
- le **Sun3/60** avec comme caractéristiques techniques :
  - un processeur 68030 cadencé à 16 MHz,
  - une puissance de 3 MIPS,
  - une mémoire de 8 Mo,
  - une interface Ethernet qui rattache la station au réseau local Ethernet du centre,
  - deux ports séries dont l'un assure la connexion à une imprimante laser Dataproducts.

La configuration comporte en outre :

- un disque de 600 Mo,
- un écran 19 pouces monochrome,
- un lecteur de cassette de 60 Mo.

Le système d'exploitation utilisé est UNIX (SunOS 4.0.3), il supporte une Interface Utilisateur Graphique à fenêtres (SunView) offrant un environnement multitâches avec un certain nombre de facilités d'utilisation de la machine.

- le **PS2/50**

processeur 80286  
1 Mo de mémoire vive,  
80 Mo de disque dur,  
un écran VGA couleur,  
raccordé au réseau par une carte Ethernet 3C503,  
avec IBMDOS ver 3.30 comme système d'exploitation.

- le **MacIIsi**

processeur 68000  
1 Mo de mémoire vive,  
40 Mo de disque dur,  
raccordé au réseau par une carte Ethernet.

- le **terminal Wyse** connecté au SPARC

Seront aussi utilisés les matériels extérieurs :

- le **PC-AT** du centre de documentation,  
processeur 80286 cadencé à 12 MHz,  
1 Mo de mémoire vive,  
40 Mo de disque dur,  
un écran VGA couleur,  
raccordé au réseau par une carte ETHERNET,  
avec MS-DOS ver 3.30 comme système d'exploitation.

- la **machine du département informatique de l'ENSUT** reliée au R.I.O.

processeur 80286 cadencé à 12MHz,  
1 Mo de mémoire vive,  
40 Mo de disque dur,

un écran VGA couleur,  
raccordé au réseau par un modem asynchrone 2400 bps via RTC (5ème solution),  
avec MS-DOS ver 3.30 comme système d'exploitation.

## **1.7. L'Etude de faisabilité**

Une fois posé le cadre de l'application, les exigences fonctionnelles et les contraintes d'utilisation à respecter ainsi que l'environnement matériel et logiciel disponible, le problème de la faisabilité du projet peut être abordé.

L'étude de faisabilité doit permettre de décider si le projet est techniquement réalisable. Il faut notamment décider de la façon dont sera reconnu l'expéditeur du message et comment les questions seront soumises à TEXTO.

### **1.7.1. La reconnaissance de l'expéditeur du message**

Il est possible de retrouver à partir de l'analyse de l'entête d'un fichier envoyé par la messagerie électronique d'UNIX l'adresse électronique de l'expéditeur du message. Cette possibilité déjà exploitée par les autres info-serveurs a été reprise pour le projet.

### **1.7.2. Le langage LOGOTEL**

Les bases de données ORSTOM sont gérées par le logiciel TEXTO qui est accompagné d'un langage de programmation (LOGOTEL), utilisé pour l'interrogation.

LOGOTEL permet d'écrire des programmes d'interface avec le logiciel TEXTO, affranchissant ainsi l'utilisateur de la connaissance des commandes TEXTO.

Dans le cas de l'info-serveur sur la base HORIZON, "l'utilisateur" sera remplacé par un module qui soumet les requêtes à la base documentaire. Ce module est développé en langage C.

L'étude de la documentation disponible sur le langage LOGOTEL montre que le dialogue se fait par l'intermédiaire du clavier/écran, ce qui est bien adapté aux applications interactives mais absolument pas aux applications différées.

Par ailleurs, il n'existe pas de variable d'environnement, qui permette à une application LOGOTEL de contrôler le résultat d'un ordre TEXTO qu'elle a activée.

Ainsi en cas d'erreur lors de l'exécution d'une commande TEXTO, s'inscrit sur le fichier de sortie (en général l'écran) des messages du genre "pardon ?", "Vous n'avez pas choisi votre dcp<sup>10</sup>", "Pas de champs implicite dans le dcp", "question ambiguë" ... puis la main est redonnée à l'utilisateur.

Ces messages très parlant lors d'une exécution interactive ne peuvent pas être facilement exploités quand il s'agit de traitement batch.

Il est néanmoins possible d'envoyer des ordres en re-dirigeant l'entrée standard sur un fichier.

Il est aussi possible sous TEXTO de re-diriger les réponses à une question sur un fichier ASCII.

### 1.7.3. Le contrôle des requêtes

Le manque de variable d'environnement utilisable sous LOGOTEL incite à une réflexion sur le contrôle des requêtes.

L'étude de LOGOTEL montre qu'il n'est pas possible de contrôler a fortiori le résultat de la soumission d'une question, car en cas d'erreur sur la question, un message d'erreur est écrit sur le fichier de sortie.

Il faut donc être sûre a priori (avant soumission) que ce résultat sera positif, c'est à dire ne soumettre à TEXTO que des questions compréhensibles par lui et qui donnent un nombre de réponses pertinent et exploitable.

## 1.8. Les caractéristiques de la solution retenue

L'étude de faisabilité a permis de dégager certaines caractéristiques de la solution.

- \*Le développement sera effectué à l'aide de plusieurs langages de programmation qui sont:
- le **langage Shell** pour développer le module de reconnaissance de l'expéditeur d'un fichier mail. Ce module est disponible, il reste à l'adapter à l'application.
  - le **langage C** pour la plupart des traitements afin de bénéficier de la souplesse et de la puissance de ce langage.

---

<sup>10</sup>**DoCumentParametre** : enregistrement spécial d'un fichier spécial "paratext" permettant à TEXTO d'identifier la structure du fichier en cours.

- le langage LOGOTEL pour la partie soumission des questions à TEXTO la récupération et la mise en forme des réponses.

\*Un module de vérification stricte des requêtes à soumettre à TEXTO.

### 1.8.1. Le contrôle des requêtes

L'étude du contrôle des requêtes est une des étapes les plus importantes du projet. Un choix a été fait entre deux démarches :

#### 1ère démarche :

- recenser de façon exhaustive les questions les plus souvent posées aux documentalistes par les chercheurs et le public,
- écrire un module de contrôle de requête rigide n'acceptant que ce genre de questions.

#### 2ème démarche :

- déterminer une grammaire la plus proche possible de celle utilisée par TEXTO pour son contrôle des équations de recherche,
- utiliser les outils disponibles sous UNIX pour écrire un module de contrôle des questions selon la grammaire spécifiée.

La première démarche présente un certain nombre de difficultés :

- le recensement ne peut pas être exhaustif,
- le temps de développement est très long,
- la maintenance d'un tel produit est très difficile;

La 2ème démarche présente de nombreux avantages :

- L'utilisation des produits de développement sous UNIX, LEX et YACC permet la réalisation d'un logiciel standard. Les délais de réalisation sont rapides et le produit réalisé est facilement maintenable.
- En outre, l'utilisation de ces outils offre l'occasion de mettre en pratique les connaissances théoriques acquises sur les types de grammaires LALR. C'est le type de grammaire accepté par YACC. Il faut pouvoir spécifier une grammaire correcte, comprendre le fonctionnement du code généré pour être capable de le modifier éventuellement.

La deuxième démarche a donc été adoptée.

Les caractéristiques du contrôle de requêtes sont :

- d'autoriser sur les champs interrogeables de la base de donnée l'ensemble des formes syntaxiques de questions acceptées par TEXTO avec en plus quelques facilités (non permises par TEXTO) :
- la liberté pour l'utilisateur d'utiliser indifféremment majuscules et minuscules pour l'écriture des questions, de mettre un nombre quelconque de caractère de séparation entre les mots utilisés,
- l'autorisation d'un vocabulaire assez libre avec tolérance aux fautes : correction de certaines erreurs orthographiques au niveau de l'écriture des noms de champs,
- le renvoi à l'utilisateur des questions incorrectes avec marquage de l'emplacement où l'erreur a été détectée.

### **1.8.2. Le moment de prise en compte de la demande de l'utilisateur**

La demande de l'utilisateur est prise en compte de façon différée essentiellement à cause des moyens de communication qu'elle emprunte : transport sur réseau avec protocole UUCP. Mais une fois arrivée au lieu de destination, elle peut être prise immédiatement en compte sous UNIX et traitée.

Sous UNIX, il est en effet possible de soumettre un job par l'intermédiaire du mail (/etc/aliases) avec le message comme point d'entrée du programme. A l'arrivée d'un message l'exécutable est lancé avec en entrée le message en question.

L'utilisation de cette possibilité permet de garantir un délai de réponse assez court à l'utilisateur. C'est cette solution qui a d'abord été choisie.

Cependant lors de la phase de tests, il est apparu que si le logiciel marche correctement lorsque la demande provient d'un utilisateur communiquant avec la machine de support de l'application en mode terminal, il est impossible d'obtenir une réponse de TEXTO lorsque l'on utilise le protocole UUCP.

Cela a posé un gros problème car le R.I.O est basé sur la solution UUCP.

A la suite de nombreux tests et après examen attentif de la documentation TEXTO, il est apparu que le logiciel TEXTO avant de s'exécuter vérifie son environnement et contrôle

notamment le terminal sur lequel il doit s'exécuter. S'il n'en reconnaît aucun, l'exécution échoue.

Il a été envisagé d'essayer de contourner la difficulté en simulant à l'intention de TEXTO la présence d'un terminal fictif.

Cela semble possible sous UNIX grâce à sa vision uniforme des entrées/sorties, véritable aubaine largement exploitée par l'info-serveur sur la base HORIZON.

Cependant, cette solution n'a pas été retenue car il n'a pas été prouvé que le mauvais fonctionnement de TEXTO provient exclusivement du contrôle de la présence d'un terminal.

Il est de plus apparu que les versions de TEXTO acquises par l'ORSTOM sont à nombre d'utilisateurs limité ( 3 à Dakar, 5 pour le site central). L'exécution de TEXTO par un 4ème (ou 6ème) utilisateur échoue.

Ce nouveau problème a définitivement écarté l'option de prise en compte immédiate de la demande dès qu'elle arrive à destination.

L'option finale utilise une autre possibilité d'UNIX, l'exécution à moments prédéterminés d'application. Ceci est faisable par l'utilisation d'une table de lancement de programmes à moment déterminés : la "crontab". Une crontab est un fichier spécial situé dans le répertoire /var/spool/cron/crontabs. Elle contient les noms des exécutable à lancer à des dates fixées. Le processus permanent cron se charge de scruter les crontabs et d'activer au moment voulu les programmes.

Dans la solution retenue, les demandes des utilisateurs reçues par un site sont conservées dans une boîte aux lettres spéciale qui est examinée en dehors des heures de travail des documentalistes. Tous les nouveaux messages sont traités par ordre d'arrivée et les réponses adéquates envoyées à l'utilisateur.

Il est intéressant de noter que cette solution "batch" est acceptée par TEXTO.

### **1.8.3. Le choix du site où sont contrôlés les requêtes**

Il aurait été intéressant de n'envoyer sur le réseau que les requêtes syntaxiquement correctes et donc d'effectuer sur la machine de départ l'analyse syntaxique.

Cependant l'étude du mail sous UNIX montre qu'un message dont l'adresse de destination est inconnue mais valide est envoyé sur le réseau; le sendmail suppose que le destinataire ne fait pas partie du site local. Il est donc impossible d'intercepter des messages à destination d'info-serveurs installés sur des bases ORSTOM distantes, d'où impossibilité d'analyse locale.

Les requêtes sont donc vérifiées au lieu de destination, et seulement lorsque la comptabilité lance l'info-serveur. C'est une des faiblesses de l'application car un utilisateur peut envoyer de mauvaises requêtes qu'il suppose correctes et devra payer la réception d'un fichier lui indiquant simplement (cas extrême) que toutes les questions envoyées sont incorrectes.

Une amélioration peut être apportée en analysant le fichier de question dès son arrivée au site (cela ne nécessite pas d'utilisation de TEXTO). Un message d'accusé de réception est renvoyé à l'utilisateur avec si nécessaire les mauvaises requêtes. Cela fera l'objet d'un développement ultérieur.

## **1.9. Le planning prévisionnel**

L'objet de ce paragraphe est de présenter la façon dont a été planifiée le temps du projet.

Le réalisation du produit devra être achevée fin avril 1993 pour pouvoir organiser au moins un stage de formation au produit courant mai.

Le logiciel à réaliser comportant des sous-systèmes de complexités différentes, c'est le modèle de "la spirale" qui a été choisi pour le cycle de développement. Ce modèle permet le développement des phases en parallèle tout en gardant les étapes du modèle waterfall classique : spécifications, conception, tests.

	1ère sem	2ème sem	3ème sem	4ème sem	5ème sem	6ème sem	7ème sem	8ème sem	9ème sem	10èm sem	11èm sem	12èm sem	13èm sem	14èm sem	15èm sem	16èm sem	17èm sem	18èm semai	19èm sem	
analyse	étude de TEXTO / LOGOTEL étude de l'environneme technique		analyse documentation analyse syntaxe des questions sous TEXTO		Etude de LEX Etude de YACC Etude des grammaire LALR Etude de TEXTO (mode ques)					Analyse du mail et du sendmail				Etude des outils de debuggage sous UNIX						
spéc.			Spécification de l'interface C / TEXTO		Spécification de la grammaire TEXTO		Spécification de la grammaire de l'infoserveur					Spécification de l'architecture globale de l'infoserveur								
concep.				module interface			Conception et compilation des fichiers LEX et YACC			Conception du module d'interrogation						Conception de l'architecture globale				
codage				module interface						Codage et test du module d'interrogation							Codage et tests des programmes		Tests de performances Debuggage	
tests												Tests de la grammaire de l'infoserveur								

## 2. Le cahier des spécifications

Le Cahier des Charges a permis de préciser quels sont les objectifs du projet et l'utilité du produit demandé.

On a défini :

- le type de l'application souhaitée : c'est un logiciel temps différé.
- le type de matériel utilisé : des stations de travail UNIX avec un environnement réseau (réseau local et réseau longue distance).
- les besoins fonctionnels demandés : transporter des requêtes vers une base documentaire, sélectionner les références correspondantes, réexpédier à l'utilisateur ces références mises en forme.
- les aptitudes du logiciels : fiabilité, adaptabilité, maintenabilité.

Les spécifications vont permettre de décrire les caractéristiques externes du système à construire et de recenser les fonctions du futur logiciel. C'est à cette étape qu'on identifie les grands blocs du projet.

### 2.1. Le contexte du développement

Cette application qui doit s'intégrer dans le système informatique ORSTOM est destinée à être implantée sur des stations de travail avec un environnement UNIX.

Cet environnement UNIX comprend en plus du système d'exploitation, un interpréteur de commandes et un grand nombre d'utilitaires : outils de mise au point des programmes, outils pour l'écriture de compilateurs, outils de messagerie électronique.

La réalisation de l'info-serveur sur la base HORIZON a bénéficiée de l'appui de quelques uns de ces outils.

L'objet des paragraphes suivants est de présenter les outils UNIX qui ont été utilisés dans la phase de spécification et de conception. Ces outils sont :

- le mail
- le sendmail
- LEX
- YACC

### 2.1.1. Le mail

Le mail est une commande utilisateur d'UNIX, appartenant au protocole UUCP, permettant à un utilisateur de composer, d'envoyer et de recevoir des messages électroniques.

Lors de l'envoi d'un message, le mail récupère le texte à envoyer et y ajoute des informations de service permettant l'identification du message, de son expéditeur et de son destinataire. Il passe ensuite le message formaté au "sendmail" chargé de router le message.

Lors de la réception d'un message, "sendmail" délivre le message en le donnant au programme mail qui ajoute ce message dans la boîte aux lettres de l'utilisateur concerné. La boîte aux lettres est un fichier qui par défaut porte le nom de login de l'utilisateur et se trouve sous le répertoire /var/spool/mail. L'ajout d'un message dans la boîte aux lettres se fait avec un protocole de verrouillage pour éviter les problèmes de mises à jour simultanées. Le mail notifie à l'utilisateur qu'un message est arrivé grâce à un démon local de notification de message /usr/etc/in.comsat, à condition que la variable biff soit positionnée à y dans le fichier de démarrage ".login".

Le mail permet également d'exécuter des applications avec en entrée des messages grâce à la commande pipe.

### 2.1.2. Le sendmail

le sendmail est une commande de gestion de messagerie du système qui permet d'envoyer un message à un ou plusieurs utilisateurs, de router le message hors du site (sur réseau longue distance) si nécessaire. sendmail réalise le service Internet de "forwarding" (faire suivre) quand nécessaire pour délivrer le message au bon endroit.

sendmail peut aussi router les messages vers d'autres serveurs connus du réseau local. La liste des serveurs pouvant directement recevoir les messages est conservée dans le fichier /usr/lib/mailhosts.

En général, le sendmail est appelé par un programme comme mail, il délivre le message pré-formaté par mail au destinataire requis et avertit le destinataire pour la délivrance du message ou insère le message en file d'attente en vue d'une transmission par réseau.

Après avoir vérifié s'il existe des options, les noms des récipiendaires sont récupérés de la ligne de commande. Une liste de récipiendaire est créée. Chaque récipiendaire est contrôlé à l'aide des informations du fichier de configuration /etc/sendmail.cf pour déterminer :

- la syntaxe du nom du réseau,
- l'alias
- la topologie du réseau
- d'autres informations ( de forwarding par exemple)

Les alias sont remplacés à cette étape et inclus dans la liste d'adressage grâce au fichier /etc/aliases.

Les récipiendaires locaux sont contrôlés ainsi que la syntaxe d'écriture des récipiendaires éloignés. Mais les contrôles détaillés des noms de sites distants sont retardés jusqu'à la délivrance.

### **2.1.3. LEX**

LEX est un générateur de programme d'analyse lexicale des flots de caractères en entrée.

Le programmeur spécifie des expressions régulières en entrée de LEX. LEX écrit le code reconnaissant ces expressions dans un flot d'entrée et partitionne le flot d'entrée en chaînes correspondants aux expressions trouvées.

Si le programmeur a spécifié un traitement spécifique pour une expression régulière quelconque en associant au fichier LEX un fragment de programme, le fragment correspondant sera exécuté pour chaque expression reconnue du flot d'entrée.

Un fichier LEX est une table d'expression régulières et de fragments de programme correspondant. Cette table est traduite en un programme qui lit en entrée un flot d'entrée, le copie sur un flot de sortie et partitionne l'entrée en chaîne correspondant aux expressions régulières données.

### **2.1.4. YACC**

YACC est un outil général pour l'écriture des compilateurs.

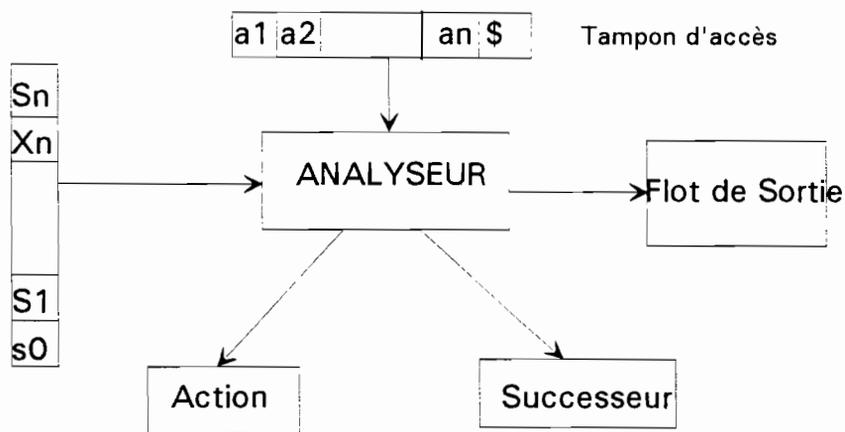
Le programmeur prépare une spécification de la syntaxe à respecter par les fichiers en entrée, ce qui inclut:

- les règles qui décrivent la structure de l'entrée,
- le code à exécuter quand une règle est reconnue,
- un sous-programme pour lire l'entrée.

YACC accepte une classe de spécification assez large : les grammaires LALR avec des règles non ambiguës.

Les grammaires LALR (Look Ahead-Left to right scanning of the input - Right most derivation in reverse) sont un des trois types de grammaires LR(k) : Left to right scanning of the input - Right most derivation in reverse. k désigne le nombre de symboles de prévision (unités lexicales) dont on tient compte à un instant donné pour décider d'une action.

Une grammaire de type LR(k) suit un algorithme d'analyse tel que le montre le schéma ci-dessous:



Une table d'analyse permet de décider à partir d'un état  $i$  donné et d'un symbole de prévision en entrée quelle va être l'action à effectuer et quel va être le nouvel état  $j$  (successeur).

Ces tables d'analyse sont difficiles (longues) à construire à la main. D'où l'intérêt d'utiliser YACC qui les élabore selon la grammaire spécifiée.

A partir des spécifications du programmeur, YACC génère une fonction qui contrôle la structure du flot d'entrée. Cette fonction appelée un parser appelle un sous-programme fournit par le programmeur (l'analyseur Lexical) pour piocher les items (les unités Lexicales) du flot d'entrée. Ces unités Lexicales sont comparées avec ce que le parser s'attend à avoir en entrée, selon la grammaire spécifiée. Si il y a correspondance, le parser continue le traitement jusqu'à la fin de la règle spécifiée ou jusqu'à ce qu'il n'y ait plus correspondance (il y a alors erreur).

Quand une règle a été reconnue, le code adéquat est exécuté.

YACC génère ses fonctions en langage C et respecte plusieurs conventions de ce langage pour la spécification de la grammaire.

## 2.3 Les spécifications fonctionnelles

L'application remplit les fonctions suivantes :

- elle examine la boîte aux lettres de l'utilisateur fictif horizon pour y vérifier la présence de nouveaux messages.
- pour chaque nouveau message, l'adresse de l'expéditeur est trouvée et la structure du fichier envoyé est vérifiée.
- le contenu du fichier est ensuite contrôlé, avec séparation des requêtes correctes des requêtes incorrectes. Celles-ci sont renvoyées à l'utilisateur.
- les requêtes syntaxiquement correctes sont soumises au logiciel de gestion de la base de données, qui sélectionne si elles existent les références pertinentes et les met en forme suivant l'option choisie par l'expéditeur.
- les réponses obtenues sont renvoyées à l'utilisateur dans sa boîte aux lettres.

### 2.3.1. examen de la boîte aux lettres horizon

Cette fonction est réalisée à intervalles réguliers déterminés par une table d'activation périodique d'exécutable.

Elle utilise le mail pour effectuer :

- la lecture de la boîte aux lettres
- la sélection des nouveaux messages
- l'activation de la fonction de contrôle et de traitement de message avec en entrée chaque nouveau message.

*objet manipulé* : BAL horizon.

### 2.3.2. contrôle de la structure du fichier

Cette fonction retrouve l'adresse de l'expéditeur, et vérifie que la structure du fichier est correcte.

Elle active la fonction de contrôle des requêtes.

*objet manipulé* : le message en entrée

*exceptions :*

- l'adresse de l'expéditeur n'a pu être trouvée
- l'adresse trouvée est celle d'horizon.
- la structure du fichier est incorrecte.

### **2.3.3. contrôle syntaxique des requêtes**

Cette fonction examine le contenu du message en entrée, en vérifiant la nature de chaque ligne.

Si une ligne donnée n'est pas une option, elle est considérée comme une question et est analysée suivant la grammaire spécifiée à l'aide de LEX et YACC.

La fonction active pour les questions correctes la fonction de soumission des questions.

*objets manipulés :* les lignes du message en entrée.

*exception :* la structure du fichier est incorrecte.

### **2.3.4. soumission des requêtes correctes**

Cette fonction active le logiciel de gestion de la base documentaire et soumet les questions correctes à l'interrogation. Elle récupère les réponses si elles existent et les met en forme dans un fichier ASCII.

*objet manipulé :* le fichier de question syntaxiquement correctes

*exception :* il n'y a pas de question à soumettre

### **2.3.5. envoi des réponses à l'utilisateur**

Cette fonction envoie tous les messages de l'info-serveur à l'utilisateur en utilisant le sendmail.

*objets manipulés:*

- fichier à transmettre
- adresse de l'utilisateur

*exception :* pas d'adresse valide

## 2.4 Les spécifications d'interface : l'interface C/TEXT0

Le logiciel TEXT0, logiciel de gestion de la base documentaire, ne peut traiter les informations que si elles proviennent du clavier ou d'un fichier TEXT0.

Il y a trois types de fichiers TEXT0:

- les fichiers index, utiles pour l'interrogation,
- les fichiers catalogues, (paratext)
- les fichiers documentaires.

Un *fichier catalogue* est un fichier d'exploitation spécial de TEXT0. Il est structuré en enregistrements qui représentent chacun la structure interne des fichiers manipulés par TEXT0. Sans fichier catalogue, TEXT0 ne peut pas travailler sur un fichier. Inversement, tout fichier dont la structure n'est pas contenue dans le fichier catalogue ne peut être exploité.

Un *fichier documentaire* est un fichier de données utilisateur. Il est constitué d'enregistrements dont la structure doit être déclarée dans le fichier catalogue pour pouvoir manipuler le fichier. Chaque enregistrement est référencé par un numéro qui peut être automatiquement attribué par TEXT0.

C'est ce type de fichier qu'utilise le module d'interface C/TEXT0 pour transmettre à TEXT0 les questions à soumettre.

Une commande TEXT0 d'ajout d'enregistrement est écrite dans un fichier ASCII avec la liste des requêtes à soumettre à TEXT0, puis le logiciel TEXT0 est lancé avec redirection de son fichier standard d'entrée vers le fichier ASCII.

Cela permet de créer un fichier exploitable par TEXT0 contenant toutes les requêtes à soumettre qui sont de plus automatiquement numérotées.

L'application LOGOTEL d'interrogation de la base peut alors être lancée avec en entrée le nom du fichier TEXT0 contenant les requêtes.

## 2.5. Les spécifications opérationnelles

### 2.5.1. Les facteurs humains

Une aide doit être fournie pour la formulation de la demande de consultation.

L'application doit prévoir :

- un fichier de présentation de la base HORIZON
- un fichier d'aide à la formulation des requêtes
- un fichier d'aide à l'écriture du message de requêtes

Ces fichiers doivent pouvoir être obtenus par demande à l'info-serveur.

Une documentation assez précise concernant la syntaxe des questions sous TEXTO doit être élaborée.

### 2.5.2. Les capacités

La taille des fichiers de requêtes est limitée par le mail à une centaine de Ko.

Le fichier de réponse transmis à l'utilisateur comporte au plus 100 références.

Le fichier de réponses est compressé si le nombre de références est supérieur à 50.

La capacité du module d'examen de la BAL horizon est arbitrairement fixée à 50 nouveaux messages.

### 2.5.3. La qualité

Au niveau des spécifications, la qualité a pour objet de traduire les facteurs du logiciel déterminés dans le Cahier des Charges en critères à remplir par le logiciel.

On reprend chaque facteur et on spécifie les actions à réaliser qui permettront d'obtenir un logiciel de qualité.

#### Fiabilité :

Il faut utiliser systématiquement des fichiers temporaires à noms uniques en utilisant les identificateurs de processus.

Il faut accéder à la base documentaire par la commande "ques" de TEXTO qui réduit le nombre de commandes réalisable et vérifier que la question posée ne peut pas être prise pour une commande TEXTO.

#### Adaptabilité :

Il faut paramétrer toutes les fonctions au maximum et concentrer les options spécifiques au site dans un seul fichier.

#### Maintenabilité :

L'utilisation de **LEX** et **YACC** permet une facilité de modification de la fonction du contrôle des requêtes.

Cependant la modification de certaines options traitées par **TEXTO**, comme augmenter le nombre de réponses maximal du fichier réponses entraînera une réécriture partielle du programme **LOGOTEL**.

#### Rapidité :

Utiliser tous les moyens de programmation pour que le temps de traitement soit au plus de 5 min. Le temps de réponse sera assez long à cause de la solution différée.

#### Portabilité :

L'utilisation du langage C et du Bourne Shell rendent totalement portable l'application qui peut être mise en oeuvre sur tout site UNIX où a été installé **TEXTO**.

#### Conformité :

Il faut s'assurer que logiciel correspond au Cahier des Charges.

#### Maniabilité :

L'application nécessite la connaissance de la messagerie UNIX et du logiciel **TEXTO**.

La messagerie UNIX est un standard à l'ORSTOM. Elle est largement utilisée par les chercheurs.

Le logiciel **TEXTO** est d'un emploi assez facile, proche du langage naturel. Cependant, le manuel d'utilisation doit comporter des explications claires et détaillées sur toutes les possibilités de formulation des requêtes.

#### Qualité de la recherche :

Pour l'utilisateur, l'efficacité de l'info-serveur sera liée au fait que l'application fournit bien :

- les références des documents voulus,
- toutes les références pertinentes du fonds documentaire

- rien que les références pertinentes.

La qualité de la recherche est donc fonction :

- de la pertinence,
- de l'exhaustivité,
- du bruit ou du silence de la réponse.

La pertinence dépend surtout des compétences de l'interrogateur, il doit avoir de bonnes notions de logique et connaître un peu TEXTO pour exprimer correctement ses désirs.

L'exhaustivité dépend des performances de TEXTO.

Le bruit et le silence dépendent de l'équation de recherche (la question posée), des limites techniques du logiciel d'interrogation, de la structuration de la base documentaire et surtout de la qualité de l'indexation.

L'info-serveur horizon est lui un filtre totalement transparent entre la question posée par l'utilisateur et celle soumise à TEXTO. Il n'y a donc pas de déformation due au traitement de l'info-serveur.

#### **2.5.4. Les bases de données utilisées**

##### *2.5.4.1. la base brute : HORISEN*

Le fichier contenant le stock documentaire à Dakar s'intitule HORISEN.

Il est structuré en 81 champs récapitulés dans les tableaux suivants. La première colonne des tableaux représente le code du champ, la seconde colonne est le code développé, la troisième colonne l'explication du champ et la quatrième colonne contient un exemple pour chaque champ. Lorsque des exemples n'ont pas été trouvés pour un champ donné dans la base HORISEN, cela est mentionné par l'expression "non rempli".

<b>AFA</b>	Affiliation d'auteur de l'article	Organisme auquel un auteur se rattache professionnellement Pour les auteurs ORSTOM indique le département auquel ils appartiennent	/1ORSTOM /3DES.Département Sante /4Paris /5FRA
<b>AFM</b>	Affiliation de l'auteur de la monographie	idem	/ORSTOM /3MAA. Département Milieux et Activités Agricoles UR E 4AOA /4DAKAR /5 SEN
<b>AMA</b>	Auteur moral de l'article	éditeur scientifique : collectivité ou personne responsable de l'édition d'un document du point de vue de son contenu intellectuel éditeur scientifique	/1UNICEF /4Abidjan /5CIV (ed.)
<b>AMM</b>	Auteur moral de la monographie	éditeur scientifique	/2GROUPE ECOLE SUPERIEURE DE COMMERCE LYON /3Dep. Environnement et Entreprise /4Lyon /5FRA
<b>ANI</b>	Animaux	Descripteurs des animaux dont traite le document	CRICONEMELLA CURVATA
<b>AUA</b>	Auteur de l'article	personne physique	/Briend, André
<b>AUM</b>	Auteur de la monographie	personne physique	Sasson, Albert
<b>BIBLIOG</b>	Bibliographie	Nombre de références bibliographiques de l'ouvrage	64
<b>BULL 1</b>	Bulletin 1	Code permettant de sélectionner ou non le document pour l'édition du bulletin bibliographique	1
<b>CANI</b>	Candidats animaux	Descripteurs des animaux traités, ne faisant pas partie du thésaurus utilisé (IBISCUS)	nom rempli
<b>CARTE</b>	Carte	Nombre de carte(s) avec mention d'échelle	2
<b>CDE</b>	Candidats descripteurs	Mots-clés indiquant le sujet du document, mais ne faisant pas partie du vocabulaire HORIZON et susceptibles d'y entrer	ECONOMIE INDUSTRIELLE
<b>CETH</b>	Candidats ethnie	Mots-clés indiquant les ethnies traitées par le document mais qui ne sont pas indiqués dans le thésaurus IBISCUS	PULLO
<b>CHAP</b>	Chapitre	Code thématique appartenant au plan de classement HORIZON	096INDUS
<b>CLING</b>	Candidats langue	Mots-clés indiquant les langues traitées dans le document mais qui ne font pas partie du thésaurus	non rempli

<b>CMORG</b>	Candidats micro-organisme	Mots-clés indiquant les micro-organismes traités dans le document mais qui ne font pas partie du thésaurus	PLASMIDIUM FALSIPARUM
<b>COLL</b>	Collection	mention de la collection d'une monographie	Sextant (FRA)
<b>CONFID</b>	Confidentiel	Code permettant de cacher ou non le document à l'interrogation	1
<b>CORG</b>	Candidat organisme	Organisme traité par le document mais qui ne fait pas partie de la liste des organismes utilisée dans le lexique IBISCUS	non rempli
<b>COTE</b>	Cote physique du document	Indique l'emplacement physique du document dans le centre	D TH 155
<b>CROC</b>	Candidat rocher	Descripteur des roches traitées dans le document qui n'appartiennent pas au thésaurus utilisé	non rempli
<b>CVEG</b>	Candidats végétaux	Descripteur des végétaux traités dans le document qui n'appartiennent pas au thésaurus utilisé	non rempli
<b>DA</b>	Date	Date de saisie de la notice	881123
<b>DACG</b>	Dates du congrès	Dates de début et de fin du congrès	89/11/19-20
<b>DADIPL</b>	Date diplôme	date d'obtention du diplôme	1988/09/20
<b>DES</b>	Descripteurs	Mot ou groupe de mots retenus dans le vocabulaire HORIZON et choisis parmi un ensemble de termes équivalents pour représenter sans ambiguïté une notion contenue dans le document	THEORIE ECONOMIQUE
<b>DICT</b>	Dictionnaire	Précision supplémentaire dans le cas où l'ouvrage est un dictionnaire	non rempli
<b>DIFF</b>	Diffusion	Lorsque le document est confidentiel, le code diffusion permet de préciser quel public est autorisé à accéder au document	6B
<b>DIPL</b>	Diplôme	Intitulé du diplôme	Th. 3ème cycle : Agron. Protection des cultures
<b>DP</b>	Date de publication	Date d'édition de la publication	1993
<b>ED</b>	Editeur	éditeur commercial	Economica
<b>ETH</b>	Ethnies	Mots-clés du thésaurus indiquant les ethnies traitées par le document	HAALPULAAR
<b>GEO</b>	Localisation géographique	Descripteurs géographiques provenant du thésaurus d'IBISCUS (Base de données de la coopération française)	SENEGAL FLEUVE

<b>GRAPH</b>	Graphique	Nombre de graphique(s)	62
<b>ILL</b>	Illustration	Nombre de dessins, schémas	1
<b>INDBIB</b>	Indication Bibliographique	Code de saisie	K
<b>INV</b>	Inventaire	Numéro d'inventaire	D 06125
<b>ISBN</b>	International Standard Book Number	Numéro international normalisé du livre. Identifie sur le plan international chaque livre ou chaque édition d'un livre d'un éditeur déterminé.	2-7178-0590-7
<b>ISSN</b>	International Standard Serial Number	Numéro international normalisé d'une publication en série. Identifie sur le plan international le titre-clé d'une publication en série.	0037-9085
<b>JT</b>	Titre de la revue		Monographs in Evolutionary Biology (USA)
<b>LAN</b>	Langue	langue dans laquelle est écrit le document	FRE
<b>LANR</b>	Langue résumé	langue des résumés existant dans le document	ENG
<b>LED</b>	Lieu d'édition	ville d'édition du document	Vevey (CHE)
<b>LING</b>	Langue	Descripteur(s) de langue(s) traitée(s) dans le document provient de la liste francis (CNRS)	FRANCAIS
<b>LOC</b>	Localisation géographique	région(s), village(s), quartier(s) n'appartenant à aucun thésaurus	MOUTOUL CUVETTE
<b>LOCG</b>	Localisation congrès	ville et pays où ont eu lieu le congrès	Lyon
<b>MATDAC</b>	Matériel d'accompagnement	Document complémentaire à l'ouvrage destiné à être utilisé avec celui-ci	1 disquette 5*1/4
<b>MIC</b>	Microfiche	Code utilisé à Bondy. Indique si le document existe sur microfiche	non rempli
<b>MIORG</b>	Micro-organisme(s)	Descripteurs des micro-organismes traités dans le document. Provient du thésaurus IBISCUS	GLOMUS MOSSAE
<b>NBEX</b>	Nombre d'exemplaires	Nombre d'exemplaires possédés par le centre	1
<b>NIVBIB</b>	Niveau bibliographique	Précision du niveau bibliographique de l'ouvrage	M

<b>NOBRA</b>	Numéro du programme ou de la convention		N° TS 2A 0017 F CD
<b>NODIPL</b>	Numéro diplôme	Numéro d'ordre du diplôme	26
<b>NOEDI</b>	Nouvelle Edition	Mention s'il s'agit d'une nouvelle édition ou si c'est une édition corrigée	N
<b>NOMCG</b>	Titre du congrès		Problematique et Methodologie en Economie Industrielle
<b>NOTE</b>	Note	Remarque intéressante qui ne peut être mise dans aucun champs	financé par le CNRS
<b>NUM</b>	Numéro de la notice	Numérotation automatique de Texto. Format : 6 chiffres. Pas : 1	572
<b>NUMCG</b>	Numéro congrès	Numéro du congrès	4
<b>NUMER</b>	Numéro	Numéro de la collection ou de la revue	3
<b>OPERAT</b>	Opération	Programme dans lequel a été réalisé l'étude	Sénégal ORSTOM/ CEE Grand programme Eau et Santé
<b>ORG</b>	Organisme(s)	Descripteurs des organismes traités dans le document. Répertoriés dans le lexique des organismes IBISCUS	OCDE
<b>PAGE</b>	Page	Nombre de pages	192p
<b>PAYSCG</b>	Pays du congrès	Code ISO du pays où s'est tenu le congrès.	FRA
<b>PERS</b>	Personnes	Noms de personnes célèbres dont parle le document.	UMAR AL-HAJJ
<b>PHOTO</b>	Photo	Nombre de photos	18 n.b.
<b>RES</b>	Résumé	Texte d'environ 100 mots décrivant de façon objective le contenu du document. Peut être absent Peut être écrit par l'auteur du document.	
<b>ROC</b>	Roche(s)	Descripteurs des roches traités par le document tirés du thésaurus IBISCUS	non rempli
<b>SOL</b>	Sol(s)	Descripteurs des sols traités par le document	SOL SULFATE ACIDE

<b>TABL</b>	Tableaux	Nombre de tableaux	4
<b>TIFRAA</b>	Titre français de l'article	Traduction du titre original en français de l'ouvrage spécifique (article ou chapitre)	Regulation d'une croissance foetale normale : aspects nutritionnels
<b>TIFRAM</b>	Titre français de la monographie	Traduction du titre original du document générique (contenant le document spécifique) en français.	Besoins nutritionnels et évaluation d'une croissance normale
<b>TITANA</b>	Titre Anglais de l'Article	Traduction du titre du document spécifique (article) en anglais	Studies on malaria in a forest-savane mosaic area of Central Africa
<b>TITANM</b>	Titre Anglais de la monographie	Traduction du titre du document générique (monographie ou document contenant l'article) en anglais	New data about <i>Criconemella curvata</i>
<b>TITORA</b>	Titre Original de l'Article	Titre original du document spécifique (article d'une revue ou chapitre d'une monographie)	Normal foetal growth regulation : nutritional aspects
<b>TITORM</b>	Titre Original de la monographie	Titre original du document générique (monographie ou revue)	Economie industrielle : problématique et méthodologie
<b>TYPBIB</b>	Type bibliographique	signale la forme physique du document	B
<b>UNIV</b>	Université	Université de délivrance du diplôme	Université C.A.D. de Dakar
<b>VEG</b>	Végétaux	Descripteurs des végétaux traités par le document	ACACIA SENEGALENSIS
<b>VERSION</b>	Version	Indique quand nécessaire si le document est provisoire, partiel	Pr.
<b>VOL</b>	Volume	Numéro de volume d'une revue	3
<b>VOLO</b>	Volume Ouvrage	Mention indiquant si l'ouvrage est en plusieurs volumes	non rempli

LES INDEX

Le fichier HORISEN est associé à 9 fichiers index :

IINV, ICOTE, IAU, ICHAP, IDES, IGEO, ITI, IDP, IBASE qui sont chacun lié à des champs.

NOM INDEX	NOM DE CHAMP(S) ASSOCIE(S)
IINV	INV
ICOTE	COTE
IAU	AUM, AUA
IAMM	AMM, AMA
ICHAP	CHAP
ITI	TI . . .
IDP	DP
IGEO	GEO, LOC
IDES	DES, CDE, ETH, CETH, LING, CLING, ORG, CORG, PERS, VEG, CVEG, ANI, CANI, MIORG, CMIORG, SOL, ROC, CROC, RES.

L'interrogation de la base par l'info-serveur est autorisée par sept champs : DP, DES, AU, CHAP, DES, GEO, TI qui sont tous indexés.

L'interrogation sera donc automatiquement effectuée par recherche dans l'index associé.

L'index du champ DES est associé à tous les champs d'indexation. Ce qui permet de les interroger simultanément.

LA TAILLE

HORISEN comporte actuellement environ 1300 enregistrements.

La taille globale du fichier documentaire est d'environ 1,2 Mo (1 321 629 octets).

C'est une petite base comparée au fichier central HORIZON qui comporte près de 40 000 références.

#### 2.5.4.2. La base d'interrogation : horisen

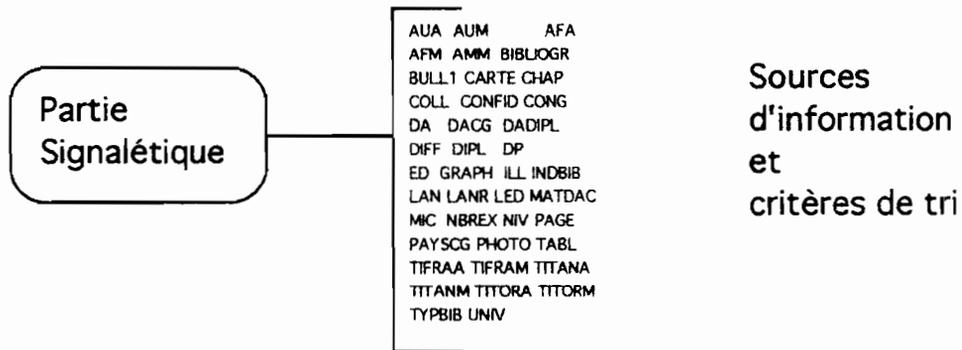
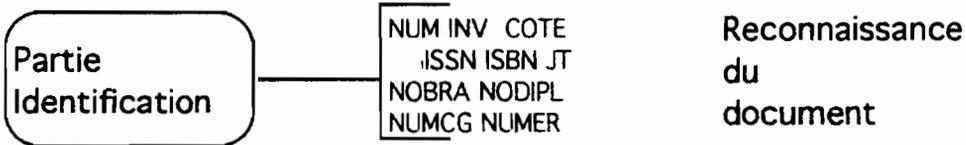
La base d'interrogation horisen est un fichier obtenu par transformation du fichier HORISEN à l'aide d'un programme logotel : format.

Le fichier horisen comporte cinq champs : NUMERO NOTICE SOURCE CHAP MOTCLE RESUME qui rassemblent le contenu des champs élémentaires de la base HORISEN en 4 grands ensembles d'information du plan documentaire comme le montre le schéma l'UNITE DOCUMENTAIRE de la page suivante.

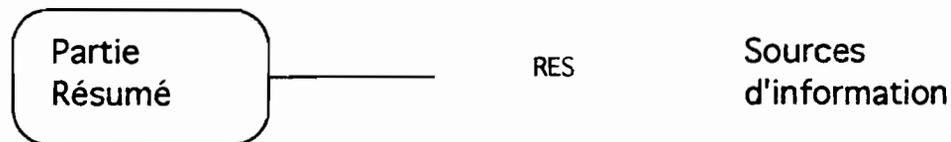
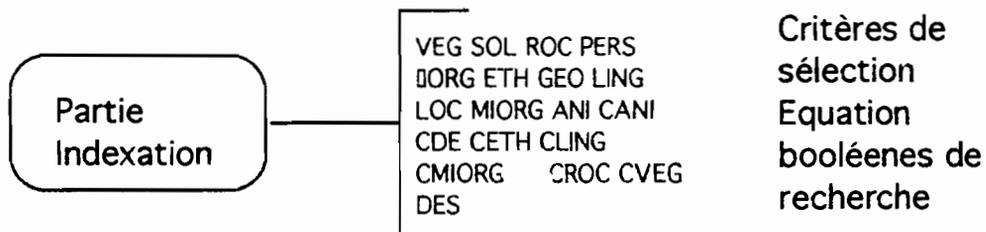
champs	plan documentaire
NUMERO NOTICE	partie identification
SOURCE CHAP	partie signalétique
MOTCLE	partie indexation
RESUME	partie résumé

C'est ce fichier qui sera utilisé pour l'interrogation.

Éléments signalétiques



Analyse du contenu



L' UNITE DOCUMENTAIRE

### 2.5.5. Les standards : présentation des notices

Les références bibliographiques peuvent être présentées suivant une des deux façons exposées ci-dessous :

#### référence développée :

C'est celle qui est transmise par défaut.

Elle comporte la notice bibliographique qui donne tous les renseignements signalétiques, les mots-clés utilisés pour indexer le document et le résumé s'il existe.

HORIZON - 62473 - Delaunay, Daniel - Démographie et systèmes agraires : un exemple d'inférence statique dans un système d'information géographique - IN : Mullon, Christian (ed.) - le transfert d'échelle dans un système ORSTOM, Paris (FRA), 1991, pp. 255-263, cart. - (Colloques et Séminaires (FRA) - 4. Séminfor 4 : Quatrième Séminaire Informatique de l'ORSTOM, Brest, FRA - (F A31958/1; B CB52 35/2)

020MATHO1

DEMOGRAPHIE; POPULATION RURALE; SYSTEME AGRAIRE; SYSTEME D'INFORMATION GEOGRAPHIQUE; VARIATION SPATIALE; CARTOGRAPHIE TRANSFERT D'ECHELLE / EQUATEUR

#### référence limitée :

Elle ne comporte que la notice bibliographique. Cette notice est présentée en s'inspirant de la norme AFNOR NF Z 44-005 relatives aux références bibliographiques.

HORIZON - 62473 - Delaunay, Daniel - Démographie et systèmes agraires : un exemple d'inférence statique dans un système d'information géographique - IN : Mullon, Christian (ed.) - le transfert d'échelle ORSTOM, Paris (FRA), 1991, pp. 255-263, cart. - (Colloques et Séminaires (FRA) - 4. Séminfor 4 : Quatrième Séminaire Informatique de l'ORSTOM. Brest. FRA - (F A31958/1; B CB52 35/2)

020MATHO1

## 3. La conception

### 3.1. Liste des modules

L'application est composée de cinq modules :

- exam\_BAL,
- infoserveur\_horizon,
- verif\_syntaxe,
- traitement\_requete,
- loginfos2.

### 3.2. Conception des modules

#### 3.2.1. exam\_BAL

##### 3.2.1.1. Objectif du module

Ce module développé en langage C a pour objet d'examiner la Boite Aux Lettres de l'utilisateur horizon, pour en extraire les nouveaux messages (ceux qui sont arrivés depuis le dernier examen) et les faire traiter par infoserveur\_horizon.

##### 3.2.1.2. Description fonctionnelle

Lecture de la BAL horizon par la commande mail et mémorisation de son contenu dans un fichier temporaire.

Analyse du fichier temporaire. Les numéros des nouveaux messages sont enregistrés dans un tableau.

Ouverture de la BAL horizon par la commande mail et pour chaque nouveau message, appel à infoserveur\_horizon.

##### 3.2.1.3. Donnée d'entrée et vérification

le fichier /var/spool/mail/horizon :

Boite Aux Lettres horizon. Fichier spécial (ASCII) qui contient tous les messages reçus par horizon.

Tous les contrôles concernant ce fichier sont effectués par mail.

#### **3.2.1.4. Taille du module**

code : 4 086 octets

exécutable : 24 576 octets

#### **3.2.1.5. Les contraintes d'appel**

Ce module est lancé à intervalles déterminés par une table d'exécution périodique du module.

Pour s'exécuter, ce module doit avoir en argument le chemin complet de l'exécutable qui traite les nouveaux messages, c'est à dire infoserveur-horizon.

#### **3.2.1.6. Tests appropriés pour juger de la qualité du module**

Vérifier ce qui se passe lorsque :

- un nouveau message arrive pendant l'exécution du module
- il n'y a aucun nouveau message
- le nombre de nouveaux messages est supérieur à la capacité de traitement du module.

#### **3.2.1.7. Exception et traitement**

Dépassement de capacité du module : si il y a plus de 50 nouveaux messages, les messages suivant le 50ème ne sont pas traités. Ils doivent cependant toujours être considérés comme nouveaux à la prochaine exécution du module.

### 3.2.2. infoserveur\_horizon

#### 3.2.2.1. Objectif du module

Ce module développé en Shell détermine l'adresse de l'expéditeur du message. Il active les modules de vérification de syntaxe des requêtes et de traitement des requêtes. Il renvoie à l'utilisateur le résultat de ses requêtes.

#### 3.2.2.2. Algorithme

DEBUT

Recherche de l'expéditeur

si l'adresse est trouvée et adresse différente de celle de horizon

alors activation du module verif-syntaxe

si un fichier de requêtes incorrectes a été créé

alors renvoi du fichier à l'utilisateur

finsi

si un fichier de requêtes correctes a été créé

alors activation de traitement-requete

si le fichier de réponse doit être compressé

alors envoi d'un message d'avertissement à l'utilisateur

compression et envoi du fichier de réponse

sinon envoi du fichier de réponse sans compression

finsi

sinon envoi d'un message de mauvais format du fichier de requête

finsi

sinon envoi du message au gestionnaire du mail

finsi

FIN

#### 3.2.2.3. Donnée d'entrée et vérification

message : fichier spécial formaté par le mail. Structuré en entête et contenu du message.

Le contenu du message est lui-même structuré en :

requête horizon

< ligne de requête ou option >

.....

.....

< ligne de requête ou option >

fin

#### **3.2.2.4. Liste des données en sortie**

Messages de réponse :

- message d'avertissement en cas de compression du fichier de réponse
- message d'avertissement de mauvais format du fichier de requêtes
- message d'avertissement de mauvais fonctionnement d'un module appelé
- message contenant les réponses aux requêtes correctes
- message contenant les requêtes syntaxiquement incorrectes.

#### **3.2.2.5.Taille du module**

script Shell : 7989 octets

#### **3.2.2.6.Les contraintes d'appel**

Le module appelant doit fournir le message en entrée sous Shell par redirection de l'entrée, sous le mail par utilisation du filtre ( le pipe).

#### **3.2.2.7. Tests appropriés pour juger de la qualité du module**

- En cas d'exécutions simultanées du module, exécutions lancées avec en entrée des messages différents, vérifier qu'aucun fichier n'est effacé, réécrit, envoyé au mauvais destinataire, et que les réponses sont toujours pertinentes.
- vérifier que les messages de horizon ne sont pas traités
- vérifier que l'expéditeur reçoit toujours une réponse.
- que se passe-t-il si le message reçu est vide mais avec une structure correcte.

### 3.2.2.8. Traitement des exceptions

- L'adresse de l'expéditeur n'a pas été trouvée. => le message est envoyé au gestionnaire de la messagerie.
- l'adresse de l'expéditeur est celle de horizon => le message est envoyé au gestionnaire de la messagerie.
- L'exécution d'un module appelé s'est mal passée. => information envoyée au gestionnaire et à l'expéditeur.

### 3.2.3. verif\_syntaxe

#### 3.2.3.1. Objectif du module

Ce module développé en langage C a pour objet la séparation du contenu du message en fichiers de bonnes et mauvaises requêtes

#### 3.2.3.2. Algorithme

DEBUT

Vérification de la structure du contenu du message

si structure incorrecte

alors création du fichier réponse contenant un message de mauvais format

sinon lecture du message reçu jusqu'à la ligne "requête horizon"

Lecture de la ligne suivante

Tant que fin du message non atteinte ou que ligne "fin" non reconnue

si la ligne lue est une spécification d'option

alors recopier la ligne dans le fichier des bonnes requêtes

sinon s'assurer que la ligne est conforme à la grammaire spécifiée

si ligne conforme

alors recopier la ligne dans le fichier des bonnes requêtes

sinon recopier la ligne dans le fichier des mauvaises requêtes.

finsi

finsi

lire la ligne suivante

Fin Tant que

finsi

FIN

### 3.2.3.3. Description fonctionnelle

La grammaire qui doit être respectée par les requêtes est la suivante :

```

<requete> := <requetecomp><requetesimple> | <requetesimple>
<requetesimple> := '(' <requeteprim> ')' | <requeteprim>
<requeteprim> := CHAMPS '=' <expression> | CHAMPS '=' <expressionsimple> | '(' <requete> ')'
<expressionsimple> := '(' <expression> ')' | <phrase>
<phrase> := <phrase> MOT | MOT
<expression> := <expressioncomp><expressionsimple>
<expressioncomp> := <expressionsimple> TRANSEXP
<requetecomp> := <requete> TRANSREQ | '(' <requete> ')' TRANSREQ

```

Cette grammaire est contrôlée par un analyseur syntaxique générée par YACC à partir des spécifications données.

L'analyseur syntaxique communique avec un analyseur Lexical généré par LEX qui contrôle la chaîne de caractères en entrée selon le principe du tableau ci-dessous :

Expression régulière reconnue dans le fichier d'entrée	Unité Lexicale retournée à l'analyseur syntaxique.	Chaîne de caractères écrite dans le fichier de sortie
{DES}/"="	CHAMPS	DES
{DP}/"="	CHAMPS	DP
{AU}/"="	CHAMPS	AU
{GEO}/"="	CHAMPS	GEO
{CHAP}/"="	CHAMPS	CHAP
"="{ESPACE}*	"="	"="
"("	"("	"("
")"	")"	")"
{SAUF}/"(" * {MOTCLE }	TRANSREQ	SAUF
{OU}/"(" * {MOTCLE }	TRANSREQ	OU
{ET}/"(" * {MOTCLE }	TRANSREQ	ET
{SAUF}	TRANSEXP	SAUF
{OU}	TRANSEXP	OU
{ET}	TRANSEXP	ET
{CHAINE}{ESPACE}	MOT	CHAINE mise en majuscule
" "[^]" "	MOT	'tous les caractères lus'

conventions utilisées :

/ suivi par

^ excepté

\* le caractère précédent en 0 ou plusieurs fois

Les chaînes {AU}, {DES}, {DP}, {CHAP}, {GEO}, {SAUF}, {OU}, {ET} sont reconnues dans le fichier d'entrée écrites en minuscules ou en majuscules.

{CHAINE} est reconnue si il y a une suite de caractères quelconque dans le fichier d'entrée.

{ESPACE} est reconnu si il y a une ou plusieurs tabulations et/ou espaces.

#### **3.2.3.4. Donnée d'entrée et vérification**

fichier de requêtes. il y a vérification de la structure.

Il doit être structuré comme suit :

requete horizon

< ligne de requete ou option >

.....

.....

< ligne de requete ou option >

fin

#### **3.2.3.5. Liste des données en sortie**

- fichier de mauvaises requêtes
- fichier de requêtes correctes
- fichier d'avertissement de mauvaise structure du fichier

#### **3.2.3.6. Taille du module**

code : 6 150 octets

exécutable : 172 032 octets

#### **3.2.3.7. Les contraintes d'appel**

Le module appelant doit préciser les noms des fichiers en sortie en les passant en argument.

#### **3.2.3.8. tests appropriés pour juger de la qualité du module**

La grammaire doit être particulièrement testée . La vérification doit déterminer :

- quelles sont les requêtes jugées correctes,
- quelles sont les requêtes jugées incorrectes.

### **3.2.3.9. traitement des exceptions**

Le message en entrée ne peut être ouvert en lecture. => abandon du module.

### **3.2.4. traitement-requete**

#### **3.2.4.1. Objectif du module**

Ce module développé en langage C a pour objet de soumettre à l'interrogation les requêtes correctes et de récupérer les réponses de TEXTO en activant le module loginfos2. Il détecte les options spécifiées par l'utilisateur et les transmet à loginfos2 de façon compréhensible pour celui-ci.

### 3.2.4.2. Algorithme

DEBUT

si le fichier en entré est de structure incorrecte

alors création du fichier réponse contenant une aide pour l'écriture d'un fichier de requêtes  
positionner le code de retour à -3.

sinon lecture du message reçu jusqu'à la ligne "requete horizon"

Lire la ligne suivante

Tant que fin du message non atteinte ou que ligne "fin" non reconnue

si la ligne lue est une spécification d'option

alors positionner la variable correspondante

sinon ajouter la ligne dans un fichier temporaire de requêtes à soumettre

finsi

Lire la ligne suivante

Fin tant que

si une option de demande de fichier d'aide est positionnée

alors copier le fichier demandé dans le fichier réponse

finsi

si le nombre de requêtes à soumettre est non nul

alors aller dans le répertoire de travail de TEXTO

créer les fichiers d'interface avec TEXTO

activer le module loginfos2

copier le fichier de réponses de TEXTO dans le fichier de réponses

revenir dans le répertoire courant

sinon le signaler dans le fichier de réponses

finsi

si l'option de compression est positionnée

ou que le nombre de réponses est supérieures à 50

alors positionner le code de retour à 4.

sinon positionner le code de retour à 0.

finsi

finsi

FIN

### 3.2.4.3. Liste des données d'entrée et vérification

fichier de bonne requête structuré comme suit :

"requete horizon"

< liste de requêtes correctes et d'option >  
"fin"

Il y a vérification de la structure.

#### **3.2.4.4. Donnée en sortie**

fichier de réponses contenant le résultat de la soumission des requêtes et structuré selon les options précisées.

#### **3.2.4.5. Taille du module**

code : 8 790 octets  
exécutable : 155 648 octets

#### **3.2.4.6. Les contraintes d'appel**

Le module appelant doit transmettre par la ligne d'argument le nom du fichier réponse où les réponses seront écrites, et le répertoire de travail où sont créés les fichiers temporaires.

#### **3.2.4.7. Tests appropriés pour juger de la qualité du module**

L'interface entre l'environnement C et l'environnement LOGOTEL doit être particulièrement vérifiée.

Il faut s'assurer que processus TEXTO se termine quelque soit le cas.

#### **3.2.4.8. Traitement des exceptions**

Le processus TEXTO boucle à cause d'un signal parasite reçu. => arrêt du processus TEXTO et du module traitement\_requete.

### **3.2.5. loginfos2**

#### **3.2.5.1. Objectif du module**

Ce module développé en langage LOGOTEL a pour objet de soumettre à l'interrogation un ensemble de requêtes supposées syntaxiquement correctes, de vérifier pour chacune d'elles

le nombre de réponses obtenues, et de mettre en forme ces réponses selon l'option spécifiée par l'utilisateur.

### 3.2.5.2.Algorithme

DEBUT

Lire le fichier d'entrée pour charger dans des variables de travail des renseignements nécessaires à l'exécution fournis par le module appelant.

Initialisation de l'environnement de travail de TEXTO.

Lire le fichier de question pour chargement d'une question en mémoire

Tant que la fin du fichier des questions n'est pas atteinte

soumettre la question à la base documentaire

recupérer le nombre de réponses

si le nombre de réponses est nul

alors le signaler dans le fichier de réponse

sinon si le nombre de réponse est  $> 100$

alors le signaler dans le fichier de réponse

sinon si la somme du nombre de réponses actuelle

et du nombre de réponse déjà copiées dans le fichier de réponses

est  $> 100$

alors le signaler dans le fichier de réponses

sinon récupérer les réponse et les mettre en forme selon le format spécifié

si le nombre de réponses du fichier = 100

alors le signaler dans le fichier de réponses

Tant que la fin du fichier de questions n'est pas atteinte

Lire la question suivante et la copier dans le fichier de réponses

Fin Tant que

finsi

finsi

finsi

finsi

lire le fichier de question pour chargement de la question suivante

Fin tant que

Ecrire le nombre de réponses du fichier de réponses dans le fichier spécifié.

FIN

### 3.2.5.3. Liste des données d'entrée et vérification

- fichier de question : fichier spécial TEXTO contenant les questions à soumettre.
- documents paramètres :  
 ficpar qui reflète la structure du fichier documentaire avec association des index  
 ficparh1, structure du fichier documentaire sans association d'index  
 ficparh, structure du fichier de travail contenant les références pertinentes à mettre en forme  
 ficparh2, structure du fichier de réponses sans association d'index  
 pquestio, structure du fichier de questions
- documents d'édition :  
 edform, utilisé pour la mise en forme avec mots-clés et résumé ;  
 ednotice, utilisé pour la mise en forme sans mot-clé et sans résumé;  
 edbrut, utilisé pour la présentation brut des références
- l'option de mise en forme : nombre entier
- chemin complet du fichier de réponses
- chemin complet du fichier devant contenir le nombre de réponse du fichier de réponses
- chemin complet du fichier de données
- nom des fichiers temporaires de mise en forme

Pas de vérification.

### 3.2.5.4. Liste des données en sortie

- fichier de réponses mises en forme selon l'option spécifiée
- fichier contenant le nombre de réponses du fichier de réponses

### 3.2.5.5. Taille du module

script LOGOTEL : 17 264 octets

remarque : un programme LOGOTEL est un fichier documentaire. Il est donc accompagné d'un fichier paramètre, d'un fichier de verrouillage et de fichier d'index (vide).

### 3.2.5.6. Les contraintes d'appel

Le module s'attend à recevoir toutes les données d'entrée à partir du clavier. Elles doivent être écrites dans un ordre précis dans un fichier ASCII. L'entrée standard du module est ensuite re-dirigée vers le fichier ASCII.

### 3.2.5.7. Tests appropriés pour juger de la qualité du module

Vérifier ce qui se passe lorsque :

- le nombre de références d'une question est supérieur à 100,
- il n'y a aucune réponse pertinente correspondant à la question,
- le nombre de références du fichier de réponses est égal à 100,
- le nombre du fichier de question est supérieur ou égal à cent.

### 3.2.5.8. Traitement des exceptions

\*TEXTO ne peut pas interpréter la question (faute de syntaxe ou question ambiguë). => On passe à la question suivante après avoir signalé le fait dans le fichier de réponse. Il n'y a pas de variable d'environnement, mais il existe un fichier historique des questions sous TEXTO qui mémorise toutes les questions posées et traitées. L'exploitation de ce fichier permet un contrôle des questions posées.

\*En cas d'erreur non prévues par le module, TEXTO boucle en envoyant indéfiniment sur le fichier de sortie le même message ("pardon?"). => il faut tuer le processus.

\*Si le nombre de questions du fichier de requêtes est supérieur ou égal à 100, les questions de rang égal et supérieur à 100 ne sont pas traitées.

## 4. L'interface

L'étude de l'interface a pour objet de décrire quels sont les éléments de communication des logiciels communiquant ainsi que la façon dont la communication va se dérouler.

### 4.1. Présentation de l'interface : modules et logiciels mis en jeux

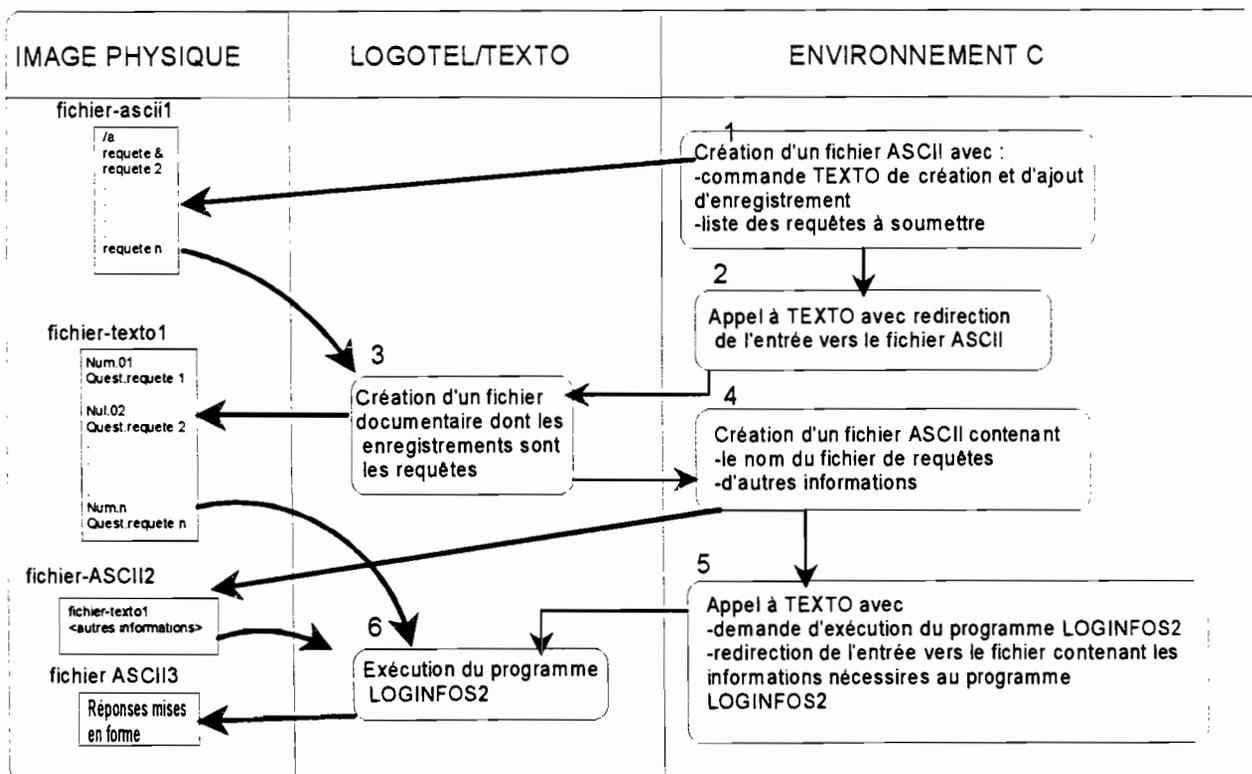
Les modules qui communiquent sont traitement-requete et loginfos2. Le premier est développé en langage C, le second en langage LOGOTEL.

traitement-requete doit transmettre à loginfos2 les questions à soumettre les noms des fichiers résultats et l'option de mise en forme des réponses.

loginfos2 doit fournir à traitement-requete les références mises en forme et leur nombre

### 4.2. Description de la structure de l'interface

Le fonctionnement de l'interface entre traitement-requete (langage C) et loginfos2 (LOGOTEL) est présentée sur le schéma ci-après.



## 5. Les tests

### 5.1. Objectifs

Les tests réalisés ont pour objectif de vérifier que les propriétés du logiciel et les fonctions qu'il doit remplir telles qu'elles ont été définies dans le cahier des charges sont effectives:

Il faut vérifier que le produit réalisé fonctionne correctement et tendre vers le "zero default" de Bill Crosby (qualiticien),

Il faut s'assurer que le produit réalisé est conforme au Cahier des Charges.

### 5.2. La vérification du produit

La vérification du produit a consisté à effectuer des tests de façon "bottom-up" sur chaque module développé. Les petits modules ont été testés séparément puis les tests d'intégration ont permis de vérifier qu'assemblés, les modules testés fonctionnent correctement.

La plupart des modules a été testée selon la méthode de la boîte blanche. Il s'agit d'effectuer des tests structurels pour repérer tous les chemins possibles dans l'algorithme. On donne ensuite des valeurs aux variables telles que tous les chemins possibles soient parcourus.

Des tests de performance ont été effectués avec les commandes **prof**, **tcov** et **time** sous UNIX.

L' exécutable `/usr/5bin/time nom-exec` permet d'obtenir le temps d'exécution total de l'exécutable nom-exec, le temps passé en mode utilisateur, et le temps passé en mode système.

La commande `prof nom-exec` génère un rapport qui indique quels sont les temps d'exécution et le nombre d'appel de chaque symbole externe.

La commande `tcov` génère un à partir d'un exécutable de programme C un listing du code commenté. Celui-ci contient notamment pour chaque fonction le nombre d'appels qui ont été exécutés.

Trois messages de requêtes ont été constitués et envoyés à chaque module. Le premier message comporte 10 requêtes, le second 50 requêtes et le troisième 100 requêtes.

Les paragraphes suivant présentent les résultats des tests effectués regroupés par modules.

Pour chaque test de fonctionnement, on présente un tableau avec l'événement testé, le résultat attendu et le résultat obtenu.

Pour chaque test de performance, on présente les temps d'exécution total, en mode utilisateur, et en mode système.

Les tests de performances ont été réalisés à des heures où le système informatique n'est pas sollicité, entre 18h30 et 19h30. Les ressources n'étant plus partagées, on obtient des résultats qui reflètent réellement les temps d'exécution.

### 5.2.1. exam-BAL

#### Test de fonctionnement

événement produit	résultat attendu	résultat obtenu
un nouveau message arrive pendant l'exécution du module	le message n'est pas traité mais est conservé	le message n'est pas traité et reste nouveau pour une prise en compte ultérieure lors de la prochaine exécution
aucun nouveau message	pas de traitement sur les anciens messages	sortie du module sans traitement
nombre de nouveaux messages supérieur à 50	Seuls les 50 premiers messages sont traités	Les 50 premiers messages sont traités. Le reste est pris en compte lors d'une exécution ultérieure

#### Tests de performances

nombre de nouveaux messages / nombre de messages total	temps réel d'exécution (en sec)	temps mode utilisateur (en sec)	temps mode système (en sec)
10/73	2.5	1.0	0.8
25/73	2.9	1.1	0.8
50/73	4.7	1.0	1.2

### 5.2.2. infoserveur-horizon

#### *Test de fonctionnement*

événement produit	résultat attendu	résultat obtenu
message sans adresse	envoi d'un message au gestionnaire	message envoyé au gestionnaire
exécution "simultanée" avec deux fichiers de requêtes provenant d'utilisateurs différents	pas de mélange de message	pas de mélange de fichier. Chaque utilisateur reçoit des données cohérentes.
le message reçu a une structure correcte mais pas de contenu	envoi d'un message à l'expéditeur	le message est retourné à l'utilisateur avec une remarque explicative
le message reçu a une structure incorrecte	envoi d'un message à l'expéditeur	le message est retourné à l'utilisateur avec une remarque explicative
l'adresse de l'expéditeur est celle de horizon	le message est ignoré	envoi du message au gestionnaire

### 5.2.3. verif-syntaxe

C'est ce module qui a fait l'objet de la période de test la plus longue. Il a été testé selon la méthode de la boîte noire : un ensemble de questions le plus exhaustif possible a été créé et soumis à verif-syntaxe avec les résultats suivants ( seul un échantillon représentatif des questions testées est fourni) :

REQUETE EN ENTREE	RES. ATTENDU	RESULTAT OBTENU	RESULTAT OBTENU
		REQUETE CORRECTE	REQUETES INCORRECTES
DES =ECONOMIE *	correct	DES =ECONOMIE *	
DES =*ECONOMIE	incorrect	DES =*ECONOMIE	
MOT-CLE =TRAVAIL	correct	DES =TRAVAIL	
personne =Beauvoir	incorrect		PERSONNE erreur =BEAUVOIR
Lieu de Travail =Dakar	incorrect		LIEU erreur DE TRAVAIL =DAKAR
localite =SINDOUF	correct	GEO =SINDOUF	
chapitre =M189	correct	CHAP =M189	
Auteur =Bergson	correct	AU =BERGSON	
Lieudetravail =Dakar	correct	GEO =DAKAR	
date =jeudi	correct	DP =JEUDI	
dp =jeude	correct	DP =JEUDE	
dportation =vendredis	correct	DP =VENDRE DIS	
mot =@vide	incorrect	DES =@VIDE	
description =^\$3	incorrect	DES =^\$3	
auteur =Simone de Beauvoir'	correct	AU =Simone de Beauvoir'	
au =<T*	correct	AU =<T*	
geostationnaire =vrai	correct	GEO =VRAI	
geologie =mot	correct	GEO =MOT	
champ =auteur	incorrect		CHAMP erreur =AUTEUR

mot_cle =villa	incorrect	DES =VILLA	
village =sen >*	incorrect		VILLAGEerreur=SEN >*
date = >1990	correct	DP =>1990	
geo =senegal et dakar	correct	GEO =SENEGAL ET DAKAR	
geo ='SENEGAL ET DAKAR'	correct	GEO ='SENEGAL ET DAKAR'	
auteur =Birago ou Branchon	correct	AU =BIRAGO OU BRANCHON	
chap =M160 sauf M189	correct	CHAP =M160 SAUF M189	
des =arbre ou poisson ou oiseau	correct	DES =ARBRE OU POISSON OU OISEAU	
des =arbre et poisson et oiseau	correct	DES =ARBRE ET POISSON ET OISEAU	
des =arbre sauf poisson sauf oiseau	correct	DES =ARBRE SAUF POISSON SAUF OISEAU	
des =arbre et poisson ou oiseau	correct	DES =ARBRE ET POISSON OU OISEAU	
des =(arbre et poisson) ou oiseau	correct	DES =(ARBRE ET POISSON) OU OISEAU	
des =foret ou fleuve sauf riviere	correct	DES =FORET OU FLEUVE SAUF RIVIERE	
geo =dakar et (pikine ou guediewaye)	correct	GEO =DAKAR ET (PIKINE OU GUEDEWAYE)	
geo =(dakar ou saint-louis	incorrect		GEO =(DAKAR OU SAINT-LOUISerreur
au =(diatta ou diata) sauf diata b*	correct	AU =(DIATTA OU DIATA) SAUF DIATA B*	
des =dakar et ((ville sauf commune) ou maison sauf baraque))	incorrect		DES =DAKAR ET ((VILLE SAUF COMMUNE) OU MAISON SAUF BARAQUE))erreur
des =senegal ou cameroun) sauf politique	incorrect		DES =SENEGAL OU CAMEROUN)erreur SAUF POLITIQUE

des=(senegal ou cameroun) sauf (politique et (geograpjie ou (localite sauf village)))	correct	DES=(SENEGAL OU CAMEROUN) SAUF (POLITIQUE ET (GEOGRAPJIE OU (LOCALITE SAUF VILLAGE)))	
(des=banque et argent) sauf (geo=senegal)	correct	(DES=BANQUE ET ARGENT) SAUF (GEO=SENEGAL)	
des=banque et argent sauf geo=senegal	correct	DES=BANQUE ET ARGENT SAUF GEO=SENEGAL	
(geo=dakar et (pikine ou guediewaye)) et ((des=banlieu*) ou (dp=1990))	incorrect		(GEO=DAKAR ET (PIKINE OU GUEDEWAYE)) ET ((DES=BANLIEU*) OU (DP=1990))erreur
geo=dakar et (pikine ou guediewaye) et (des=banlieu ou dp=1990)	correct	GEO=DAKAR ET (PIKINE OU GUEDEWAYE) ET (DES=BANLIEU OU DP=1990)	
au=(traore ou dp=1990)	incorrect		AU=(TRAORE OU erreurDP=1990)
au=(traore ou) et dp=1990	incorrect		AU=(TRAORE OU )erreur ET DP=1990
au='traore ou' et dp=1990	correct	AU='traore ou' ET DP=1990	
geo=((senegal et ferlo) ou (mali et sahel)) sauf (casamance et sine.saloum)) et des=(argent ou banque*) sauf (FMI ou Banque Mondiale) et dp =>1990 sauf (1991 et 1993)	correct	GEO=((SENEGAL ET FERLO) OU (MALI ET SAHEL)) SAUF (CASAMANCE ET SINE.SALOUM)) ET DES=(ARGENT OU BANQUE*) SAUF (FMI OU BANQUE MONDIALE) ETDP=>1990 SAUF (1991 ET 1993)	

Le résultat des tests est globalement positif. Seules les requêtes qui comportent des mots commençant par un caractère spécial sont acceptées alors qu'elles ne devraient pas l'être. Il faut interdire que le début d'une chaîne commence par \* ou par . sinon, la recherche sera faite séquentiellement dans le fichier documentaire.

Les requêtes ambiguës sont aussi acceptées, car l'analyse est syntaxique et non sémantique.

#### Tests de performances

nombre de requêtes	temps réel d'exécution (en sec)	temps mode utilisateur (en sec)	temps mode système (en sec)
10	1.3	0.0	0.5
50	4.9	0.2	1.1
100	5.6	0.3	1.7

#### 5.2.4. traitement-requete

##### Test de fonctionnement

événement produit	résultat attendu	résultat obtenu
le processus TEXTO boucle	terminaison du programme avec code -1	terminaison du programme avec code -1 au bout de 5 min.
le fichier de requêtes est de format non conforme	envoi d'un fichier d'aide à la construction du fichier	le fichier de réponse en sortie contient un fichier d'aide
le fichier de requête est vide (mais de format conforme)	envoi d'un message d'explication	le fichier de réponse en sortie contient une note explicative

##### Tests de performances

nombre de requêtes	temps réel d'exécution (en sec)	temps mode utilisateur (en sec)	temps mode système (en sec)
10	27.9	8.8	4.4
50	1:29.8	24.4	7.8
100	2:47.8	47.5	12

La lenteur du module s'explique par le fait qu'il attend que le processus loginfos2 se termine pour continuer son exécution.

Si le temps réel d'exécution varie du simple au double lorsqu'on passe de 50 à 100 requêtes, le temps d'utilisation de l'unité centrale reste sensiblement la même.

La différence temps d'utilisation - (temps mode utilisateur + temps mode système ) donne le temps de mise en attente du processus. Il est de 1:48.3 pour 100 requêtes soit 64,5% du temps total. Il est de 57.6 sec. pour 50 requêtes soit 64.1% du temps total.

### 5.2.5. loginfos2

#### Test de fonctionnement

événement produit	résultat attendu	résultat obtenu
plus de 100 questions sont en entrée	Seules les 99 premières sont traitées	Seules les 99 premières questions sont traitées.
le nombre de références correspondant à une requête est supérieur à 100	Cette requête n'est pas traitée	Cette requête n'est pas traitée. Elle est copiée dans le fichier de réponses avec une note d'explication
il n'y a aucune référence correspondant à la question posée	Une note mentionne le fait dans le fichier de réponse	Une note mentionne le fait dans le fichier de réponse
le nombre de références du fichier de réponse est égal à 100	Les autres requêtes du fichier de question ne sont pas traitées	Les requêtes du fichier de question sont copiées dans le fichier de réponse sans traitement
le document paramètre transmis ne correspond pas au fichier à interroger	TEXTO boucle	TEXTO boucle
Une des réponses attendues du fichier d'entrée n'est pas obtenue	TEXTO boucle	TEXTO boucle

#### Tests de performances

nombre de requêtes	temps réel d'exécution (en sec)	temps mode utilisateur (en sec)	temps mode système (en sec)
10	23.1	8.5	2.6
50	1:26.5	24.2	6.0
100	2:49.3	47.1	9.7

Lorsque tout ce passe bien, le temps de traitement de TEXTO est inférieur à 3 min. dans le cas extrême où il y a 100 questions à traiter. On peut donc fixer la limite du temps d'attente du module de traitement-requete à 5 min.

### 5.3. La validation du produit

L'étape de validation a pour objet de décider si le produit est accepté ou pas.

Dans le cas de ce projet, les tests de validation ont pris deux formes :

- un stage de formation de deux jours a été organisé le 18 et le 19 mai 1993 pendant lesquels quelques futurs utilisateurs au nombre de 6, ont utilisé le produit.

- une démonstration a été effectuée devant le responsable du secteur Documentation, M. Stjepanovic et le responsable du réseau CDO Outremer (Centres de Documentation ORSTOM) et de l'appui informatique à la Documentation, M. Rossi le 15 juin 1993.

M. Rossi a installé une nouvelle version de TEXTO, la version 4.2.2. Cette version offre de nouvelles possibilités parmi lesquelles celle de pouvoir interroger la base documentaire sur un index de base. Il n'est alors pas nécessaire de donner un nom de champ.

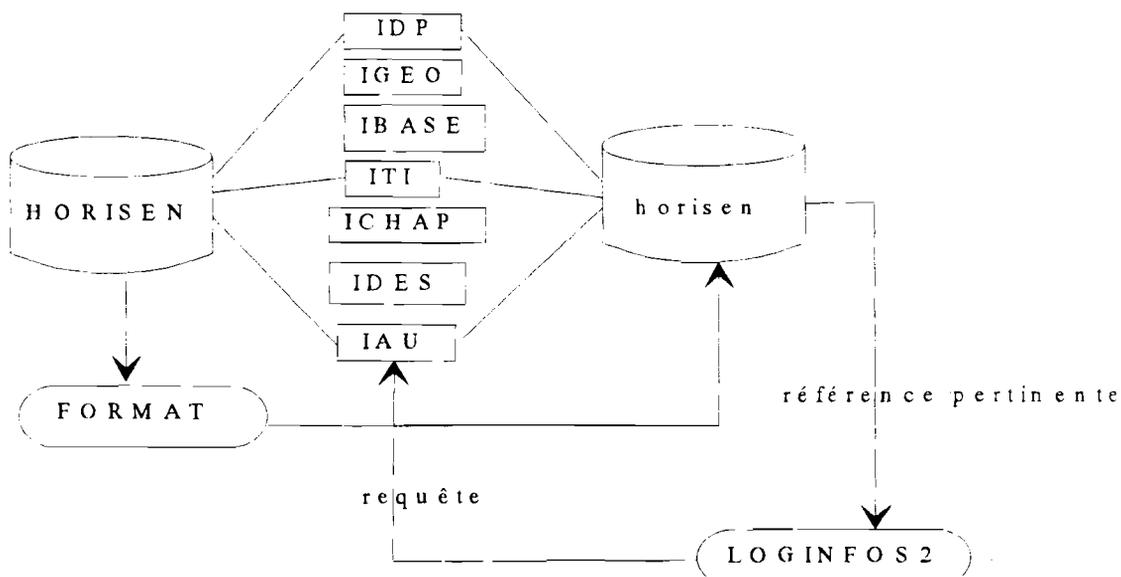
Cette possibilité devra être prise en compte.

Le responsable de l'Informatique à la Documentation ORSTOM a aussi demandé qu'une modification de fonctionnement soit apportée au module loginfos2.

Il existe déjà une application qui met en forme les enregistrements de la base brute HORISEN en les transformant en références bibliographiques dans la base horisen. L'info-serveur n'a pas à utiliser la base au format de saisie, mais plutôt la base au format d'interrogation.

Ce changement a été pris en compte, ce qui simplifie beaucoup le module loginfos2. Les tests de performance présentés ont été effectués sur le module modifié.

Vue des fichiers utilisés par LOGINFOS 2 modifié



## 6. Le manuel d'utilisation

### 6.1. Caractéristiques essentielles de l'info-serveur sur la base HORIZON

Pour l'utilisateur, les caractéristiques essentielles de l'info-serveur sur la base HORIZON sont :

-le mode DIFFERE

-la possibilité de consulter n'importe quelle base documentaire ORSTOM

-la possibilité d'obtenir le résultat de la consultation avec une mise en forme choisie par l'utilisateur

-la tolérance aux fautes sur les noms des champs

-la possibilité d'interroger la base sur 5 champs différents.

### 6.2. Comment fonctionne l'info-serveur sur la base HORIZON

L'info-serveur sur la base HORIZON fonctionne de la manière suivante :

il recueille un ensemble de questions provenant d'un utilisateur dans un fichier de questions structuré de façon particulière.

il contrôle chaque question du fichier de requête et renvoie à l'utilisateur les questions syntaxiquement incorrectes.

il soumet chaque question syntaxiquement correcte à l'interrogation et recueille pour chacune les références pertinentes.

il retourne à l'utilisateur l'ensemble des références pertinentes.

Nous allons voir maintenant :

Quelle est la structure d'un fichier de questions

Comment envoyer à la base documentaire de son choix un tel fichier.

### 6.3. Structure d'un fichier de requête

Un fichier de requêtes destiné à l'info-serveur sur la base HORIZON est structuré de la façon suivante :

```
requête horizon
liste de
requêtes
et de
lignes d'option
fin
```

### 6.4. Comment interroger la base documentaire ORSTOM de son choix

Vous pouvez interroger la base documentaire ORSTOM de votre choix en envoyant à cette base un fichier de requêtes structuré comme indiqué au paragraphe 6.3

La liste des bases documentaires ORSTOM accessibles est disponible par la commande annuaire horizon.

Exemple :

Pour obtenir toutes les références d'ouvrages traitant de banque contenues dans la base documentaire de Dakar, vous envoyez un mail à [horizon@dakar.orstom.sn](mailto:horizon@dakar.orstom.sn) comme ceci :

```
mail horizon@dakar.orstom.sn
Subject :<retour chariot>
Cc :<retour chariot>
requete horizon
DES=BANQUE
fin
```

Nous allons maintenant nous intéresser au contenu du fichier de requêtes.

## 6.5. Majuscules et minuscules

L'emploi des majuscules et des minuscules est indifférent pour l'élaboration du fichier de requêtes.

## 6.6. Le contenu du fichier de requêtes

Le contenu du fichier de requêtes est un ensemble de lignes dont chacune est :

- soit une requête
- soit une option

## 6.7. Les options possibles

Il y a 4 options possibles :

- aide fichier
- aide question
- compresse
- format limite

### 6.7.1. L'option "aide fichier"

Cette option permet d'obtenir dans le fichier de réponse un texte d'aide à l'écriture du fichier de requêtes.

### 6.7.2. L'option aide question

Cette option permet d'obtenir dans le fichier de réponse un texte d'aide à l'écriture des questions.

### 6.7.3. L'option compresse

Cette option permet d'obtenir un fichier de réponses compressé.

Remarque : Le fichier de réponses est automatiquement compressé si le nombre de réponses est supérieur à 50.

#### 6.7.4. L'option format limite

Cette option permet d'obtenir des références bibliographiques sans les mots-clés et le résumés qui décrivent le contenu du document.

Si une ligne du fichier de requêtes commence par une de ces 4 expressions, elle est considérée comme une option.

Les options ne sont pas soumises à l'interrogation.

### 6.8. Les requêtes

#### 6.8.1. Les champs interrogeables

Les notices bibliographiques peuvent être retrouvées par l'interrogation de certains de leurs champs :

Nom du champ	Développé	Structure du champ	exemple
AU	Auteur	Phrase	AU=Boulvert, Yves AU=BOIVIN
DES	Descripteur	phrase ou mots-clés	DES=RECOLTE DES=UTILISATION DU SOL
DP	Date de publication	Mot	DP=1990
GEO	Géographie	Mot ou Phrase	LAC DE GUIERS SENEGAL
CHAP	Chapitre	Mot	096INDUS

#### 6.8.2. La syntaxe des questions à respecter pour formuler une question

Pour interroger la base bibliographique, il faut respecter la syntaxe de TEXTO concernant les requêtes.

Ce paragraphe présente les différents éléments syntaxiques d'une question sous TEXTO et la façon de les agencer pour obtenir une question compréhensible par TEXTO.

## 1. Les éléments syntaxiques d'une question TEXTO

### article

Le contenu d'un champs peut former un tout (ex : résumé) ou être constitué d'un ou plusieurs articles (ex : mot-clé).

Les articles sont des suites de caractères séparés entre eux par des ";". Ils autorisent la constitution d'index permettant de faciliter et d'accélérer l'interrogation.

exemple de contenu du champs DES :

TIERS MONDE; SOUS DEVELOPPEMENT; MODELE DE DEVELOPPEMENT;  
STRATEGIE DE DEVELOPPEMENT; ANALYSE

Tous les champs interrogeables de l'application sont des champs divisés en articles.

### troncature (\*)

Une troncature est un joker qui permet de masquer une chaîne de caractères inconnus. Elle équivaut à une chaîne de caractère de longueur et de contenu indéfinis.

TEXTO autorise les troncatures gauches, droites et internes.

### masque (.)

Le masque est aussi un joker mais qui ne masque qu'un caractère.

### inférieur (<) et supérieur (>)

Les symboles < et > peuvent être employés au sein d'une chaîne de caractères avec une signification de comparaison "alphabétique".

### opérateurs logiques

TEXTO autorise les opérateurs logiques "ET" "OU" "SAUF"

### parenthèses

Les parenthèses permettent de préciser des combinaisons logiques de chaînes de caractères recherchées au sein d'un champ. Elles permettent également de poser une question portant simultanément sur plusieurs champs.

## 2. La question de base

### définition

Dans TEXTO, la question de base est constituée par un article dont l'utilisateur demande la présence dans un champ. Une question de base a la forme symbolique :

$C=A$  avec :

$C$  = Nom du champs interrogé

$A$  = Article devant être présent dans le champ " $C$ " pour qu'un document donné réponde à la question.

### exemple

$GEO=SENEGAL$

Toutes les notices dont le champs GEO contient le nom SENEGAL seront sélectionnées.

## 3. extension de la question de base

La question de base offre des possibilités limitées. Un document dont le champs GEO contient "SENEGAL NORD" et non "SENEGAL" ne répondra pas à la question précédente.

L'utilisation de masques, troncatures, d'opérateurs logiques permet de transformer la question de base en question plus élaborée du type:

$C=X$

avec :

$C$  = Nom du champs interrogé

$X$  = Chaîne de caractères au sens le plus large, grâce aux troncatures, masques, <, >, ET, OU, SAUF

Dans les exemples qui suivent, les symboles suivant seront utilisés:

- $C$  = le nom d'un champs,
- $X, Y$  = chaînes de caractères,
- ... = une chaîne de caractères **quelconque** ayant un **nombre**

**de caractères quelconques.**

- x, y = caractères quelconques.

4. troncatures (\*)Utilisation

## QUESTION

Le champ "C" des documents répondant à la question contiendra une chaîne de caractères telles que :

C = X*	<=>	X...
C = X*Y	<=>	X...Y
C = *	<=>	Pour qu'un document réponde, il suffit que son champs "C" ne soit pas vide.

L'intérêt de la troncature est qu'elle permet d'élargir la question posée.

Exemples

GEO=CASAMANCE\*

Toutes les références dont le champs GEO contient des articles commençant par "CASAMANCE" répondront à cette question.

(CASAMANCE BASSE, MOYENNE, HAUTE, BASSIN, BASSE COURS D'EAU etc)

DES=FIEVRE\*DE LA VALLEE DU RIFT

Références dont le champs DES contient des articles commençant par "FIEVRE" et se terminant par "DE LA VALLEE DU RIFT".

(FIEVRE HEMORRAGIQUE DE LA VALLEE DU RIFT)

GEO=\* (Toutes les références qui contiennent **au moins un** lieu géographique)

Remarques

Dans la base documentaire, les articles sont conservés au singulier. Il vaut donc mieux poser des questions du genre DES=ENFANT\*, qui permettent de sélectionner toutes les références parlant d'enfants

Une question comme "GEO=\*DAKAR" où la troncature apparaît en début de chaîne est à éviter absolument, car elle entraîne une recherche séquentielle dans le fichier documentaire.

5. masques (.)

symbole, utilisation

Un masque est symbolisé par un **point**. Il permet de cacher un caractère dont on ne connaît pas la valeur exacte. Un point ne masque qu'**un** caractère. Pour masquer "**n**" caractères, il faudra donc utiliser "**n**" **points**.

C = .X            <=> xX

C = X.            <=> Xx

C = ..X           <=> xyX

C = X.Y           <=> X.Y

Etc...

*Combinaison avec des troncatures :*

C = .X\*            <=> xX...

C = X.Y\*Z\*        <=> XxY...Z...

L'utilisation du masque se justifie lorsque l'interrogation se fait avec des mots composés, pour éviter de perdre certaines références à cause du caractère de jonction différent.

exemple

DES=BAS.FOND pour BAS-FOND ou BAS FOND

AU=Cormier.Salem

GEO=SINE.SALOUM

remarque

TEXTO considère toujours le point comme un masque.

6. symboles < et >

< signifie **inférieur ou égal**.

> signifie **supérieur ou égal**.

Ces symboles provoquent la comparaison de chaînes de caractères traitées en VALEURS ALPHANUMERIQUES.

La comparaison provoquée par ces symboles est effectuée sur des chaînes de format identique (c'est à dire comportant le même nombre de caractères).

Utilisation

Les différents caractères sont hiérarchisés de la façon suivante (pour le système ASCII machine UNIX) :

Caractères particuliers < 0 1 ... 9 < A B ... Y Z < a b ... y z

On symbolise par (X) le format de la chaîne "X"

QUESTION Le champ "C" des documents répondant à la question contiendra une chaîne de caractères telle que :

C => X X'

Avec :

C = Y>X YX' (X') = (X) (formats de X et X' égaux)

X' > X au sens alphanumérique

C = >X. X'x

Application au champ DP

Le champs DP interrogeable (=date de publication) est un champs de format variable, structuré en année/mois/jour (9999/99/99) où le mois et le jour sont optionnels.

Si on recherche les références bibliographiques d'ouvrages qui ont été publiés à partir de 1989, en utilisant la question :

DP=>1989,

Seules seront sélectionnées les références où il n'ait fait mention ni du mois ni du jour de publication. Une référence dont le champs DP contient "1990/08" ne répondra pas. Ceci à cause de la contrainte de "format identique" de l'opérateur.

Pour une réponse pertinente, c'est la question DP=>1989\* qui devra être posée.

*définition finale de la question de base*

Dans TEXTO, l'utilisation des troncatures, des masques et des symboles < et > est possible simultanément. On peut poser des questions assez complexes du genre :

C =X.>Y\*

Dans ce qui suit, la question de base est :

$C = X$

avec :

$C$  = Nom du champ interrogé

$X$  = Chaîne de caractères pouvant être un article simple ou une chaîne complexe définie à l'aide des opérateurs :

\* . > <

### 7. opérateurs logiques

généralités

Les opérateurs logiques "ET", "OU", "SAUF", autorisés par TEXTO permettent d'associer plusieurs questions de base concernant un même champ : on parle alors de question composée.

Ils permettent également d'associer plusieurs questions composées, c'est à dire de poser une question simultanément sur plusieurs champs : on parle de question multichamps.

Il doit obligatoirement y avoir une chaîne de caractères avant et après l'opérateur logique employé.

Les opérateurs logiques doivent être isolés des chaînes de caractères recherchées par des blancs ou des parenthèses.

QUESTIONS COMPOSEES

exemples symboliques

$C = X \text{ et } Y$

$C = X \text{ OU } Y \text{ OU } Z$

$C = X \text{ OU } ( Y \text{ SAUF } Z)$

$C = ( X \text{ ET } Y ) \text{ OU } ( X \text{ ET } Z)$

exemple

DES=ECOETHOLOGIE OU ECOLOGIE

DES=VILLE\* ET ((SANTE\* SAUF MEDECINE) OU (POLITIQUE\* SAUF ECO\*))

QUESTIONS MULTICHAMPS

Pour poser des questions multichamps, il suffit de placer les différentes questions de base entre parenthèses et de les relier par des opérateurs logiques.

Les parenthèses ne sont nécessaires que si la question posée risque d'être ambiguë

exemples symboliques

(C1 = X ET Y) SAUF (C2=Z)

C1=X OU C2=Y

exemples

DES=BANQUE\* ET GEO=SENEGAL NORD

(GEO=AFRIQUE DE L'OUEST ET CAMEROUN) SAUF DP=<1989

### 8. échappement

La base locale de Dakar contient une référence qui dans son champs DES contient l'article "OUTIL ET TECHNIQUE DE L'INFORMATION"

En posant la question DES=OUTIL ET TECHNIQUE DE L'INFORMATION, la référence apparemment pertinente ne répondra pas.

En effet, TEXTO ne distingue pas lorsqu'il traite le texte de la question, le "ET" opérateur logique du "ET" appartenant au texte recherché (si ce dernier est lui aussi entouré de blancs).

Il traduit la question ci - dessus de la façon suivante : la référence pertinente contient dans le champ DES un article OUTIL et un article TECHNIQUE DE L'INFORMATION ; ce qui n'est pas le cas de la référence.

Donc lorsque la chaîne de caractères demandée contient "ET" "OU" "SAUF" entourés de blancs, afin que TEXTO ne confonde pas ces mots avec des opérateurs logiques, il faut masquer ces blancs.

DES=OUTIL.ET TECHNIQUE DE L'INFORMATION

Il existe une autre solution au problème qui consiste à employer un caractère d'échappement, la quote " ' ".

L'emploi de ce caractère permet d'enlever le sens syntaxique :

-des opérateurs ET OU SAUF;

-des parenthèses ;

Ainsi la question DES='OUTIL ET TECHNIQUE DE L'INFORMATION' devrait sélectionner la référence.

Cependant, la solution d'échappement pose le problème de l'utilisation de la quote ' comme apostrophe. (L'INFORMATION), qui devient alors un caractère spécial. Ce problème est résolu de 2 façons :

1. l'utilisation de masque
2. le doublement de la quote

Dans l'exemple précédent, les bonnes questions sont (par exemple):

DES='OUTIL ET TECHNIQUE DE L'INFORMATION'  
 ou DES='OUTIL ET TECHNIQUE DE L.INFORMATION'  
 ou DES=OUTIL.ET.TECHNIQUE DE L.INFORMATION  
 ou DES=OUTIL.ET.TECHNIQUE DE L'INFORMATION. ( ' n'est considéré comme caractère spécial que lorsqu'il suit immédiatement le signe =)

### 9. limites des questions

Les niveaux de parenthèses sous TEXTO sont selon le manuel d'utilisation de TEXTO "illimités".

Elles doivent cependant être judicieusement employées, car si un défaut de parenthèse peut entraîner l'ambiguïté d'une question, un excès peut au contraire entraîner la perte de références pertinentes.

ex : ((GEO=DAKAR ET SAINT-LOUIS)) donnera plus de réponses que (GEO=DAKAR ET SAINT-LOUIS). Ces réponses ne seront pas pertinentes.

La longueur maximale d'une question dépend de la machine et du système d'exploitation sur lesquels TEXTO est implanté. Sur le système UNIX, elle est de 100 caractères, ce qui correspond à environ une 1,25 ligne d'écran.

Nous allons maintenant nous intéresser à la réponse de l'info-serveur.

## **6.9. Les fichiers réponses de l'info-serveur**

La réponse de l'info-serveur est un message qui est déposé dans votre boîte aux lettres. Vous pouvez recevoir plusieurs sortes de messages suivant le fichier de requêtes que vous avez envoyé.

### **6.9.1. Le messages de réponses obtenues par interrogation de la base documentaire**

Ce message contient les références bibliographiques des documents qui répondent aux questions du fichier de requêtes.

Il est structuré en deux parties :

-une en-tête récapitule les questions qui ont été soumises à l'interrogation et les options prises en compte.

-le corps du message contient chaque question soumise à l'interrogation avec la réponse correspondante de la base documentaire.

### **6.9.2. Le message de retour des questions syntaxiquement incorrectes**

Les questions syntaxiquement incorrectes sont les lignes du fichiers de requêtes non reconnues comme option et qui ne suivent pas les règles définies au paragraphes 6.8.

Toutes les questions incorrectes vous sont retournées avec marque de l'endroit où l'erreur a été détectée. Cela vous permet de déterminer plus facilement l'erreur commise.

### **6.9.3. Le message d'avertissement en cas de compression du fichier de réponses**

L'info-serveur sur la base HORIZON vous envoie un avertissement dans le cas où le fichier de réponse est compressé.

Ce message d'avertissement vous rappelle quelle est la procédure de décompression d'un fichier.

#### **6.9.4. Le message d'avertissement en cas de problème survenu lors de l'exécution de l'application.**

Il peut arriver qu'un problème survienne lors du traitement de votre fichier de requêtes ; votre demande ne peut alors être satisfaite.

L'info-serveur sur la base HORIZON vous envoie dans ce cas une note indiquant que votre demande n'a pu être satisfaite.

Si vous recevez un tel message, il vous faut relancer l'application. C'est à dire reconstruire le fichier de requêtes et l'envoyer à nouveau à la base documentaire que vous voulez consulter.

### **6.10 Les messages d'erreurs**

Il existe deux sortes de messages d'erreurs :

- les messages concernant le fichier de requêtes
- les messages concernant les questions posées.

#### **6.10.1. Les messages d'erreur concernant le fichiers de requêtes.**

Si votre fichier de requêtes est mal structuré, vous recevrez le message suivant :

VOTRE FICHER DE REQUETES N'EST PAS au bon format.  
IL DOIT COMMENCER PAR requete horizon ET SE TERMINER PAR fin

Ce message est suivi par le texte que vous avez envoyé.

#### **6.10.2. Les messages concernant les questions posées**

AUCUNE REFERENCES NE CORRESPOND A VOTRE REQUETE

La question posée est trop restrictive.

Utiliser la troncature et le masque pour l'élargir.

IL N'Y A AUCUNE QUESTION A SOUMETTRE A TEXTO

Votre fichier de requête est vide

L'ENREGISTREMENT DES REFERENCES CORRESPONDANTES ENTRAÎNERAIT UN DEPASSEMENT DE LIMITE.

RELANCER L'INFO-SERVEUR POUR LE TRAITEMENT DE CETTE REQUETE.

Le nombre de références correspondant est inférieur à 100, la question peut donc être traitée.

La somme de ce nombre et du nombre de références déjà dans le fichier de réponse est supérieur à 100. On ne peut donc pas transmettre les références.

LE FICHER DE REPONSES CONTIENT 100 REFERENCES.

AUCUNE AUTRE REQUETE NE PEUT PLUS ETRE TRAITEE

Le nombre maximal de références dans un fichier de réponses est 100.

Si cette limite est atteinte et qu'il reste d'autres questions à traiter, celles-ci sont copiées dans le fichier de réponse sans traitement.

PLUS DE 100 REFERENCES CORRESPONDENT A VOTRE REQUETE,  
ELLE NE PEUT ETRE TRAITEE.

La question posée est trop large.

Utiliser les opérateurs logiques pour la préciser.

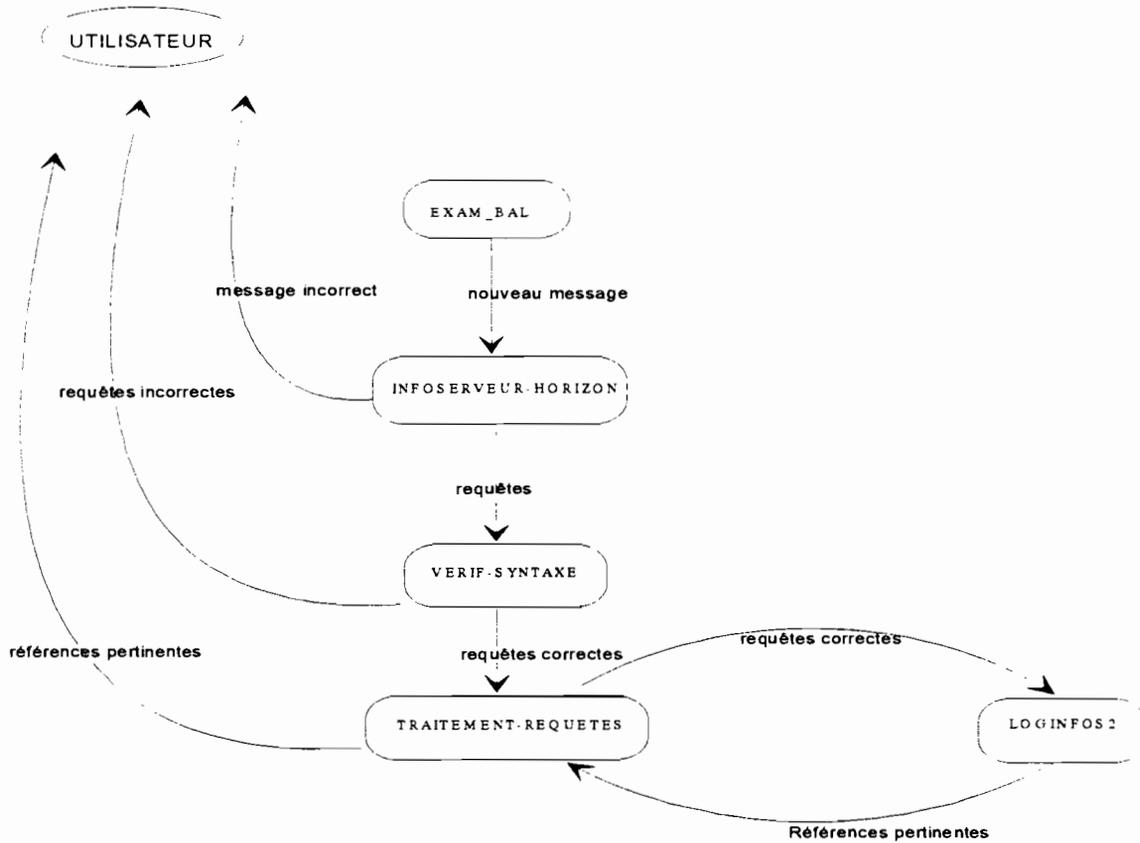
TEXTO NE COMPREND PAS CETTE REQUETE,  
ELLE EST PEUT ETRE AMBIGUE.

La question n'a pas été interprétée.

En général, cela vient d'un manque de précision. Utiliser les parenthèses.

## 7. Le manuel d'exploitation

### VUE D'ENSEMBLE



### 7.1. Configuration informatique nécessaire

La configuration informatique nécessaire est la suivante :

- système UNIX avec la librairie de fonctions de **LEX** et **YACC**
- le logiciel **TEXTO** ver. 4.2.2
- espace disque : 450 Ko
- une base documentaire **ORSTOM** avec le format décrit au paragraphe 2.5.4.

## 7.2. Messages : signification et remèdes

nom-executable : USAGE

nom-executable [liste d'argument]

L'exécutable de nom nom-executable n'a pas pu s'exécuter à cause du nombre d'arguments incorrects.

Vérifiez le fichier à partir duquel il est lancé.

nom-executable : ERREUR

Impossible d'ouvrir le fichier nom-fichier

Sortie du programme nom-executable à cause d'un problème d'ouverture de fichier.

Vérifier l'espace disque et les droits accordés sur le répertoire de travail de l'info-serveur : l'exécutable doit avoir les droits de lecture et d'écriture.

ERREUR : nom-executable

Message général en cas d'erreur système.

Cela peut être un problème de lecture ou d'écriture de fichier,

un problème de mauvais nombre d'arguments,

un problème de terminaison anormale de processus fils.

Vérifier le système de fichiers, les droits de lecture et d'écriture sur les répertoires de travail et le bon fonctionnement de TEXTO.

## 7.3. Le manuel d'installation : mise en oeuvre

7.3.1. Les fichiers sources de l'info-serveur sur la base HORIZON

Les fichiers en-tête

\*horizon.h

\*fonctions.h

### Les fichiers programmes

- \*exam\_BAL.c
- \*infoserveur\_horizon
- \*verif\_syntaxe.c
  - lex.yy.c
  - y.tab.c
- \*traitement\_requete.c
- \*loginfos2.dat
  - loginfos2.il
  - loginfos2.FRE

### Les fichiers d'aide

- \*aide\_infoserveur
- \*aide\_question

## **7.3.2. Installation de l'info-serveur sur la base HORIZON**

Pour installer l'info-serveur, réaliser les opérations suivantes:

1. Créer le compte utilisateur horizon.

Il doit disposer d'une boîte aux lettres.

2. Recopier dans le répertoire de login les fichiers sources de l'info-serveur:

- exam\_BAL.c
- infoserveur\_horizon
- verif\_syntaxe.c
- lex.yy.c
- y.tab.c
- horizon.h
- fonctions.h
- traitement\_requete.c
- loginfos2.dat
- loginfos2.il
- loginfos2.FRE
- aide\_infoserveur

## aide\_question

## 3. Editer le fichier infoserveur\_horizon et:

3.1. modifier la variable NomMailServeur pour indiquer le nom de la machine-serveur dans l'adresse électronique de l'info-serveur.

Par exemple, si l'info-serveur est installé sur la machine dakar,  
NomMailServeur="horizon@dakar.orstom.sn"

3.2. indiquer dans la variable Repertoire, le répertoire sur lequel les exécutable et les fichiers d'aide seront installés.

Par exemple,  
Repertoire="/usr2/user/horizon"

3.3. indiquer dans la variable Rep\_Tmp, le répertoire de travail où l'application pourra créer des fichiers temporaires.

Par exemple,  
Rep\_Tmp="/usr2/user/tempo"

3.4. indiquer dans la variable Gestionnaire l'utilisateur à qui seront envoyés les messages en cas d'erreur.

Par exemple,  
Gestionnaire="root"

## 4. Editer le fichier horizon.h et :

4.1. indiquer le nom de la base à interroger

Par exemple,  
basedoc[120]="/usr2/user/docdakar/horisen"

4.2. indiquer également le fichier paratext contenant les documents paramètres de la base.

Par exemple,

```
ficcat[120]="/usr2/user/docdakar/paratext"
```

4.3. indiquer le chemin complet du programme loginfos2

Par exemple,

```
prolog[80]="/usr2/user/horizon/loginfos2"
```

4.4. indiquer le chemin complet des fichiers d'aide

Par exemple,

```
fic_aide1[80]="/usr2/user/horizon/aide_infoserveur"
```

```
fic_aide2[80]="/usr2/user/horizon/aide_question"
```

5. Compiler les programmes-sources en tapant les commandes suivantes :

```
cc -o traitement_requete traitement_requete.c
```

```
cc -o verif_syntaxe verif_syntaxe.c -ll
```

```
cc -o exam_BAL exam_BAL.c
```

6. Rendre horizon propriétaire des fichiers exam\_BAL et infoserveur\_horizon:

```
chown horizon infoserveur_horizon
```

```
chown horizon exam_BAL
```

7. Etablir la périodicité de l'application

... 7.1. Créer une table des heures de lancement de exam\_BAL

Par exemple,

examen contient

```
10 7,9,12,15,18,20 * * * exam_BAL >/dev/null
```

7.2. Lancer la commande crontab pour rendre effective la périodicité

Par exemple,  
crontab examen

exam\_BAL sera exécutée tous les jours à 7h10, 9h10, 12h10, 15h10, 18h10, et 20h10.

## 8. Tester l'info-serveur

Tester l'info-serveur en envoyant le mail suivant, par exemple pour dakar:

```
# mail horizon@dakar.orstom.sn
```

```
Subject:
```

```
requete horizon
```

```
aide fichier
```

```
fin
```

L'info-serveur répondra en envoyant un message contenant l'aide pour écrire un fichier de requêtes

## Conclusion

L'application "info-serveur sur la base de données HORION" permet actuellement d'interroger en mode différé la base HORISEN de Dakar. Tous les utilisateurs du RIO peuvent ainsi savoir quelles sont les publications disponibles au centre de Dakar-Hann.

L'objectif reste de permettre une consultation de toutes les bases documentaires ORSTOM qui existent dans le monde. Il reste donc à installer le produit sur tous les sites qui possèdent une base documentaire.

Il reste aussi à régler le problème de la tarification du service.

En effet selon le principe de tarification en vigueur sur le RIO, ce sont les utilisateurs qui envoient des messages qui sont taxés pour les messages envoyés. Dans le cas de l'info-serveur sur la base HORIZON, c'est un utilisateur fictif qui envoie les réponses à l'utilisateur réel. Ce n'est pas lui qui doit être taxé mais celui qui a demandé les références.

Dans le contexte international actuel où les modes de télécommunication coûtent relativement chers, il n'est pas envisageable de procéder à des interrogations en mode conversationnel ou même d'effectuer des liaisons transcontinentales en mode IP.

La solution du mode différé est peu onéreuse et grâce aux outils disponibles sur les systèmes UNIX, il est possible d'offrir des outils d'interrogation de base de données accessibles au monde de la recherche et de l'éducation.

## BIBLIOGRAPHIE

AFNOR (Association Française de NORmalisation)

Recueil de normes françaises de documentation

2ème édition (1982)

BARES M.

Serveurs de données et réseaux télématiques

Nouvelles formes de l'information et de la communication

ed. Techniques et Documentation (Lavoisier), 1985

KERNIGHAN B. W. - RICHIE D.M.

Le langage C

ed. Masson, 1985

LUCAS H - MARTIN B - SABLET G

UNIX - Mécanismes de base, langage de commande, utilisation

ed. Eyrolles, 1984

RENAUD P

Le Réseau Informatique de l'ORSTOM (R.I.O) Besoins, solutions, performances, insertion dans l'institut, coûts, 1991

RIFLET J. M.

La programmation sous UNIX

ed. MacGRAW - HILL, 1986

TONG-TONG J.R

Les communication sous UNIX

ed. Eyrolles, 1991

YAEDOU J.

Les techniques de programmation en C

Les communications en réseaux

ed. Sybex, 1985

## **ANNEXES**

Annexe A : Internet

Annexe B : UUCP, TCP/IP

Annexe D : Principes généraux pour l'indexation des documents

Annexe E : Bordereau de saisie-indexation

Annexe F : Les programmes

## Annexe A : Internet

La construction d'un réseau interconnecté ( interconnected network - internet ) composé de différents réseaux locaux dispersés de part le monde a été un grand sujet de préoccupation de la communauté universitaire et scientifique au temps d'Arpanet.

L'idée était d'utiliser un protocole spécial appelé le protocole Internet pour masquer l'hétérogénéité physique, topologique et architecturale des réseaux rassemblés dans un unique réseau logique et virtuel. Derrière la simplicité de ce choix, un grand nombre de protocoles composent.

Sans donner aucun détail sur les protocoles Internet, il est important de souligner que de cet exemple, plusieurs solutions propriétaires ont été développées et utilisées sur différents réseaux longue distance (WAN - Wide Area Network) : EARN, BITNET, HEPNET, etc. Différents systèmes peuvent se connecter au réseau en implémentant la même pile de protocoles ou en utilisant des logiciels spécialisés pour simuler le comportement d'un réseau particulier.

Actuellement, un nouveau processus émerge de la communauté scientifique et technique, de même que dans la communauté industrielle. Il est fondé sur un paradigme réseau orienté multi-architecture et multi-protocole pour communiquer avec des utilisateurs quelque soit l'échelle, la configuration ou l'application acceptées par leurs équipements.

Le terme "Internet" sera utilisé comme terme général pour désigner l'architecture d'un réseau global . Les architectures Internet deviendront le terme général des infrastructures réseau permettant l'interconnexion des LAN (Local Area Network - réseaux locaux) et des WAN basé physiquement sur la fibre optique, la paire de cuivre, le câble coaxial, les fréquences radio (RF) ; interconnexions caractérisées par une variété de techniques de liaison de données ; ce genre d'architecture étant constituée de ponts, routeurs, et gateways constituant les blocs de l'architecture.

## Annexe B : UUCP, TCP/IP

### UUCP

Unix to Unix CoPy est une solution très utilisée qui est à la base des réseaux les plus grands comme Eunet, Usenet, Junet, SDN, Auseanet, Pacnet.

Il y a 3 principales classes de protocoles dans les réseaux UUCP :

- le protocole g qui découpe les données dans des paquets, utilise des checksums pour détecter les erreurs et retransmet quand nécessaire sur les ligne d'un réseau commuté quelconque (RTC) ;

- le protocole f qui suit la même procédure mais sur un réseau X25 (dans ce cas, UUCP n'est pas très efficace) ;

- le protocole t utilisé par les réseaux UUCP basés sur les connexions TCP/IP.

La limite de cette solution est qu'il n'est pas possible d'établir des sessions interactives entre des systèmes distants (remote login), et l'accès à des bases de données distantes n'est possible qu'en mode "batch".

Les possibilités de la messagerie et du transfert de fichiers sont considérablement restreintes par le mécanisme "store and forward" utilisé par UUCP.[2]

#### **Les fonctionnalités d'UUCP sont :**

- le transfert de fichiers
- l'exécution de commandes à distance
- la messagerie électronique
- le terminal virtuel
- l'appel d'un terminal distant

#### **Les Types de liaisons :**

UUCP supporte plusieurs types de liaisons allant d'une connexion RS-232C en asynchrone, à l'utilisation d'un réseau à commutation de paquets X25 public ou privé.

##### Liaison spécialisée.

Dans le mode de connexion directe, les ports série asynchrones (tty) de deux machines sont reliés par un simple câble ; aucun autre matériel n'est nécessaire. La distance maximum selon les standards RS-232C ou V24 est de 15m; mais dans un environnement non perturbé cette distance peut être dix fois plus grande, et même davantage. La vitesse maximale est de 19200 bits/s mais se situe à 9600bits/s en utilisation courantes. Un câble null-modem peut être nécessaire au delà d'une certaine distance lorsque la longueur de base (moins de 15 m par exemple) des câbles de chacune des installations en présence permet pas de relier physiquement les deux machines.

selon la qualité de l'environnement et de la distance, on peut être amené à utiliser des pseudo-modems ou des modems classiques. Même dans ce dernier cas, la vitesse maximum est normalement de 9600 bit/s.

##### Liaison par réseau téléphonique commuté.

Ce mode de liaison plus ouvert que le précédent permet l'établissement de communications entre différentes machines, abstraction faite de leur lieu géographique. L'élément hardware annexe indispensable est un modem. Ce modem doit être à réponse automatique avec, selon les besoins, un

dispositif de numérotation automatique (Modem Autocomposeur ou ACU Automatic Call Unit). Les vitesses de transfert sont celles en vigueur sur les différents réseaux commutés (300 bits/s, 1200bits/s, 2400 bits/s...).

Enfin dans une installation donnée, certains correspondants sont atteints par appel manuel, d'autres par appel automatique.

#### Liaison avec un réseau X25.

Les services d'UUCP peuvent être utilisés au-dessus d'X25 au moyen d'une interface appropriée sur chacune des machines communicantes. Il est bien fréquent d'avoir dans un même package, le PAD, le remote login et les services d'UUCP.

L'interface UUCP-X25 est une adaptation du protocole UUCICO au service d'X25.

Les avantages X25 par rapport au réseau téléphonique sont la vitesse de transmission (en général 9600bits/s contre 1200bits/s pour le RTC) et la fiabilité des transferts, c'est à dire une meilleure qualité de service. L'inconvénient en est la mise en place de logiciel X25 et des interfaces nécessaires ainsi que des cartes synchrones de communication. Il en résulte également un coût de communication X25, notamment dans le cas des réseaux publics.

#### Liaison par réseau local

Un réseau local peut aussi servir de support aux services standard d'UUCP à savoir le transfert de fichiers textes et binaires, l'exécution de commandes distantes, et la messagerie électronique (mail). Le type de réseau local utilisé peut être quelconque parmi les standards courants : ETHERNET, TOKEN RING, STARLAN, ETC.

### **Les Modes de fonctionnement :**

#### mode maître

Une machine en mode maître (pour une liaison donnée parmi d'autres et la machine ayant été configurée comme telle) est un noeud actif; c'est lui qui établit la connexion avec le noeud partenaire et initie les échanges. Dans le cas d'une connexion utilisant un port asynchrone tty, celui-ci doit être mis dans l'état off dans l'état off dans /etc/inittab (pas de processus getty sur le port car la machine partenaire ne se logge pas sur la machine maître) . Le statut du noeud maître est pris en considération dans les fichiers de configuration (Systems ou L.sys).

#### mode esclave

Dans ce mode de fonctionnement, la machine considérée accepte les connexions entrantes; elle ne peut donc pas appeler le noeud partenaire ni prendre l'initiative des échanges. Elle offre un service de remote login au noeud maître qui dépend du type de liaison : processus getty et procédure de login classique sur port asynchrone de connexion, serveur de connexion (de type X29) sur un réseau X25, module ou serveur tty sur un réseau local.

Bien que n'ayant pas l'initiative des transferts, le noeud esclave peut mettre en queue des requêtes (jobs) destinés au noeud maître qui viendra les chercher par interrogation (polling); ce qui suppose que le noeud esclave a été configuré au niveau du maître pour être interrogé.

#### mode bidirectionnel

Il existe un programme uugetty analogue à getty permettant à une machine d'appeler la machine partenaire et d'être appelée par celle-ci; la condition à remplir est que chacune des deux machine est un processus uugetty sur un port actif (respawn).

Cette fonctionnalité n'est pas répandue, donc pas d'utilisation courante.

### **Composition du logiciel UUCP**

UUCP est constitué de manière standard d'un ensemble de répertoires contenant les exécutables de base, les fichiers de données ou fichiers de configuration et les fichiers d'administration. Les logiciels

de communication via le réseau local ou X25 et les interfaces d'adaptation d'UUCP à ces types de liaison constituent un ensemble annexe non standard.

### Les répertoires

#### **/usr/bin**

Certains exécutables de base sont contenus dans ce répertoire; il s'agit notamment des commandes: uucp, uulog, uupick, uustat, uuto, uux, uuname, cu etc.

#### **/usr/lib/uucp**

Ce répertoire contient trois types de fichiers : programmes exécutables, scripts schell et fichiers de configuration pour uucp et cu.

- **exécutables** : uucico, uucheck, uucleanup, uusched, uuxqt, etc.
- **scripts schell** : uudemon.cleau, uudemon.hour, uudemon.poll, admin, uutry...
- **fichiers de configuration** : Devices, Dialcodes, Dialers, Permission, Systems, Sysfiles, Poll...

Selon la famille d'UNIX, Système V ou Berkeley et selon les releases dans une même famille, on a soit des fichiers supplémentaires soit des fichiers de noms différents mais ayant des fonctions équivalentes (L-devices, L.sys, L-dialcodes, USERFILE etc.)

#### **/usr/spool/uucp**

Ce répertoire contient toutes les requêtes en attente d'être traitées - ces requêtes sont rangées dans différents répertoires correspondant aux machines distantes.

En plus des répertoires des requêtes spoolées, on trouve des répertoires d'administration tels que : .Log, .Status, ainsi que des fichiers de travail tels que TM (Temporary data file).

#### **/usr/spool/locks**

Ce répertoire contient les fichiers Lock (LCK.str) créés par uucico et cu pour marquer l'occupation d'un device.

#### **/usr/spool/uucppublic**

Répertoire public d'UUCP où sont stockés certains fichiers reçus par la machine locale. En effet certaines machines distantes peuvent être astreintes à ne déposer leurs fichiers que dans ce répertoire selon les permissions qui leur sont accordés sur la machine locale.

### Les programmes

#### **uucp**

prépare les transferts vers les machines partenaires en mettant les différentes requêtes (jobs) dans une file d'attente, et en les classant selon la destination dans le répertoire /usr/spool/uucp. Les fichiers à transférer peuvent être copiés dans le spool si l'option correspondante (-c) est utilisée. Une fois le travail de préparation terminé, uucp appelle normalement le programme de transfert uucico, à moins que l'option -r de mise en queue sans transfert ait été choisie, dans ce cas, uucico est appelé autrement. Avant d'appeler uucico, uucp crée les fichiers de travail (work files C.sysnxxx) pour les requêtes et les fichiers de données (data files D.sysnxxx) pour copier éventuellement pour copier éventuellement certains fichiers avant leur transfert selon la demande de l'utilisateur.

#### **uuto**

programme analogue à uucp avec cette différence que uucp permet de choisir les répertoires sur la machine distante où doivent être transférés les fichiers, alors qu'avec uuto les fichiers sont placés dans le répertoire /usr/spool/uucppublic/receive

#### **uupick**

sert à récupérer les fichiers envoyés par uuto pour les placer dans un répertoire choisi par l'utilisateur dont le nom de login est indiqué comme sous répertoire de receive. D'autres traitements selon les options peuvent être envisagés par l'utilisateur (affichage du contenu d'un fichier, échappement vers le shell, suppression, etc)

#### **uux**

permet l'exécution de commandes sur la machine distante; comme uucp, uux crée dans le spool les fichiers de travail et éventuellement les fichiers de données, mais il crée en plus les fichiers d'exécution ( execute files X.synxxxx) contenant des informations pour cette exécution. Le transfert de ces derniers est assuré par uucico. Il peut y avoir des restrictions sur l'exécution de certaines commandes sur la machine distante.

#### **uulog**

permet d'afficher le contenu des fichiers log, cu fournissant des informations d'accès (succès, échec) au système distant.

#### **uustat**

fournit le statut des queues de requêtes, ainsi que celui des transferts initiés par uucp, uuto ou uux

#### **uucleanup**

effectue un travail de nettoyage de fichiers sous les répertoires du spool d'uucp. Ce programme est lancé dans le script Schell uudemman.cleanup démarré par cron. Les fichiers (C, D, X...) sont supprimés en fonction de leur ancienneté.

#### **uucheck**

vérifie la présence des fichiers et répertoires; une option -v permet la vérification d'erreurs dans le fichier Permissions.

#### **uutry**

script Schell permettant de relancer une commande de connexion; il fait appel au programme uucico et affiche une trace permettant de suivre les étapes nécessaires de l'établissement d'une connexion avec la machine distante. Cette trace est également placée dans un fichier sous /tmp portant le nom du système distant.

#### **uugetty**

analogue à getty mais conçu pour supporter le bidirectionnel en alternat sur une ligne tty exploitée par uucico, cu et ct. Son principe de fonctionnement repose sur les fichiers lock créés par uucico et cu: en effet uugetty ouvre la ligne de communication et se bloque en attente d'une détection de porteuse. Lorsque celle-ci est présente, uugetty tente de créer un lock sur le device; s'il y a échec, ceci signifie que le device est pris par uucico ou cu, dans ce cas uugetty se met en attente de la disparition de ce lock pour permettre un login.

#### **uucico**

processus en background, il constitue le programme de transport d'uucp. Ses fonctions peuvent se résumer ainsi :

- choix du device pour connexion
- établissement de la connexion avec la machine distante
- vérification des permissions d'accès
- transfert des fichiers (fichiers pour l'exécution distante et fichiers ordinaires textes ou binaires)
- mise à jour des fichiers log.
- indication des résultat aux utilisateurs concernés via la messagerie (mail)
- dialogue pendant toute la session avec le programme uucico de la machine distante.
- uucico est lancé par différents programmes : uucp, uux, uuto, uutry, uusched.

### **uusched**

ordonnance le travail mis en queue (requêtes spoolées par uucp, uuto, uux) pour établir l'ordre de contact des différentes machines distantes, c'est le scheduler des connexions; uusched est lancé toutes les heures par cron par l'intermédiaire du scripte schell uudemmon.hour en background.

### **uuxqt**

c'est également un processus en background; il est chargé de l'exécution distante à partir des fichiers d'exécution préparés par uux depuis la machine partenaire et transmis par uucico. uuxqt recherche les fichiers d'exécution (X.files) dans le répertoire spool; en cas d'existence d'un fichier de ce type, il vérifie l'accessibilité aux fichiers de données ainsi que l'autorisation dans le fichier Permissions d'exécution de la commande demandée. uuxqt est lancé périodiquement par cron via le script uudemmon.hour.

### **cu**

ce programme permet à un utilisateur à partir de son système de se logger à une machine distante comme un utilisateur local de celle-ci. Différents paramètres sont utilisés pour atteindre le système désiré : device, vitesse, nom du système, numéro de téléphone pour une liaison réseau téléphonique commuté. cu offre d'autres fonctionnalités telles que le transfert de fichiers ASCII, l'exécution de commandes, etc.

### **ct (call terminal)**

permet d'appeler un terminal distant; cet appel peut avoir été initié par l'utilisateur qui souhaite ne pas supporter les frais de la communication (téléphone). Pour cela, l'utilisateur peut d'abord se logger sur la machine à l'aide de son équipement (téléphone + modem à numérotation automatique), lancer la commande ct et se délogger ensuite.

## **Le protocole TCP/IP**

TCP/IP a été développé à l'origine dans le cadre du projet DARPA (Defense Advanced Research Project Agency), puis adopté et standardisé par le département de la défense Américain (DOD) en 1980 dans une série de normes RFC. Il s'est imposé ces dernières années comme le standard de fait dans la communauté universitaire, dans les réseaux locaux et environnements UNIX. La plupart des grands constructeurs ont incorporé une version de TCP/IP à leur catalogue.

TCP/IP est l'assemblage de trois protocoles indépendants: TCP, UDP et IP. Ils sont fréquemment regroupés car essentiellement utilisés ensemble. TCP/IP est prévu pour assurer la gestion complète du réseau local IEEE 802.3, dont une vue commerciale est appelée Ethernet, en apportant tous les services nécessaires aux applications réseaux.

## **Le protocole IP (Internet Protocol)**

La couche réseau du modèle OSI gère le routage ( la recherche du chemin pour l'acheminement) des informations d'un point à un autre dans un même réseau local, ou d'un réseau local à un autre. La fonction principale du service réseau permet aux utilisateurs connectés d'accéder en quelque sorte au monde extérieur.

### **Principe**

Le protocole IP fonctionne de manière très similaire au protocole réseau normalisé OSI. IP offre ainsi le service de routage de paquets entre deux systèmes à travers un ensemble de réseaux locaux. Le protocole n'a pas la notion de connexion ; il s'agit simplement de transférer des données entre deux points physiques (liaisons) de réseaux-interconnectés. Les données à transmettre proviennent de la couche TCP, et sont regroupées en datagrammes. Un datagramme est une unité de données indépendante (paquet) contenant essentiellement les informations suivantes:

- un en-tête contenant:
  - des champs divers d'indications et d'options nécessaires au routage du datagramme.
  - les adresses Internet source et destination (cf. ci-dessous),
- et les données à transmettre (contenant les TPDU et TSDU) de la couche transport).

Le Protocole IP est totalement transparent pour le développeur d'application réseau, sauf en ce qui concerne l'adresse Internet des deux correspondants.

Mode d'adressage d'IP:

IP est prévu pour fonctionner sur un réseau de type Ethernet. Sur un tel réseau, les points d'accès (stations connectées) sont référencés par une adresse, l'adresse Ethernet. Cette adresse est un code déterminé par le constructeur de la carte électronique et permettant l'accès au réseau. Chaque adresse Ethernet est unique, il n'en existe pas deux identiques au monde. Une adresse Internet est en fait une correspondance pour IP de l'adresse Ethernet. C'est le protocole ARP (Address Resolution Protocol) qui assure la correspondance. L'adresse Interne est codée sur 32 bits et comporte deux champs:

- un code indentificateur de réseau,
- un code d'identification de la machine sur le réseau.

Remarques:

- l'usage d'un routeur entre deux correspondants s'impose lorsque les champs réseau de leur adresse Internet sont différents.
- Le terme Internet désigne à la fois un protocole (IP) et nom d'un réseau international
- Le format de l'adresse est contenu dans le fichier etc/hosts.

## **Le protocole TCP (Transport Control Protocol)**

Selon les spécifications du modèle OSI, la couche transport est la charnière entre les fonctions qui traitent de la communication (gestion du réseau) et les fonctions d'exploitation (utilisation du réseau). Au-dessus de la couche transport, les caractéristiques physiques du réseau n'apparaissent plus. Les systèmes d'exploitation qui utilisent le réseau n'ont plus à se préoccuper de ces contraintes. Le service transport assure une communication de bout en bout entre deux usagers quels que soient le nombre de réseaux traversés et leur type. La couche inférieure au service transport, le service réseau, gère les ressources de communication disponibles pour l'utilisateur; la couche transport rend l'application indépendante des moyens mis en oeuvres.

Le protocole est antérieur à la normalisation ISO; il n'utilise pas tout à fait les mêmes principes. Cependant, TCP a pour rôle de rendre les mêmes services que la couche Transport du modèle OSI.

**Principe général:**

TCP s'insère entre l'équivalent de couche réseau, IP (Internetwork Protocol), et les services d'applications tels que le transfert de fichiers (réseaux interconnectés), sur lesquels on ne peut faire l'hypothèse qu'il n'y aura pas des pertes, de duplication, etc. En bref, TCP doit fournir fiable dans un environnement qui ne l'est pas.

TCP fournit un service de connexion pont à point, c'est à dire entre deux utilisateurs. Ce service garantit la livraison correcte des messages, sans duplication ni perte, quels que soient les incidents ou faute de niveau réseau. La transmission sur les connexions est full-duplex avec contrôle de flux. Le contrôle de flux est un mécanisme garantissant que l'émetteur n'envoie pas ces donnée plus vite que le récepteur ne peut les recevoir. Un mauvais contrôle de flux conduit à des pertes de données.

## **UNIX et TCP/IP**

Le système d'exploitation UNIX a d'emblée été conçu pour communiquer. Aux mécanismes de base pour la communication entre tâches, ont rapidement été ajoutés des moyens de communication en réseau. Les solutions réseaux ont été intégrées à UNIX, à mesure de leur parution, parallèlement par AT&T et l'Université de Berkeley pour aboutir à un système cohérent avec les réseaux, UNIX Système V pour AT&T, et UNIX BSD pour Berkeley. L'un des principaux protocoles intégrés à ces dernières versions, avec X25 et Xerox Network System (XNS), est le protocole TCP/IP. Le protocole Transport normalisé ISO TP4 est en voie d'intégration. Il remplacera à terme TCP/IP bien que celui-ci ait encore de longue période de prospérité devant lui.

### **Exemples de services TCP/IP sous UNIX:**

Ces services fonctionnent sur le modèle client/serveur en utilisant les protocoles UDP ou TCP: FTP (File Transfert Program). Ce service propose le transfert de fichiers UNIX vers et d'une machine distante, ainsi que l'exécution sur le serveur de quelques commandes shell relatives aux manipulations de fichiers - comme chgt de rep. ou affichage de rep. - Le processus FTP serveur exécute en local les commandes que lui a envoyé le processus FTP client et envoie les résultats à ce dernier. La connexion entre client et serveur est assurée par TCP. Le processus est le programme FTPD, le processus client est nommé FTP.

RCMD: (Remote Shell Command Execution). Ce service propose l'exécution d'une commande shell sur une machine distante. Le serveur sur la machine distante exécute la commande reçue et renvoie les résultats. Le processus client est RCMD, le processus serveur, RSHD. Le processus serveur prend fin avec la fin de l'exécution de la commande .

RLOGIN: (Remote Login). Ce service permet la connexion virtuelle du terminal local sur une machine distante. RLOGIND est le serveur, RLOGIN l'application cliente.

NFS: (Network File System). Ce service propose une extension répartie sur le réseau du système de gestion de fichiers d'UNIX. Il permet à l'utilisateur situé sur une machine connectée au réseau d'accéder à des fichiers distant de manière transparente, comme s'ils étaient sur sa propre machine. Les processus de gestion NFS sont tour à tour client ou serveur. Ils utilisent le protocole UDP.

Pour être utilisés, ces services (sauf NFS) doivent disposer de leur application serveur sur toutes les machines du réseau, puisque chaque service est susceptible d'être demandé pour n'importe quelle machine. En fait, un processus seul appelé Super Serveur INETD, est à l'écoute sur plusieurs ports (sur chaque port des services qu'il peut lancer) et lance l'exécution du serveur approprié pour une demande de connexion de chaque client.

Pour fonctionner, ces services sont répertoriés dans et utilisent des fichiers d'information, comme il est vu ci-après.

### **L'organisation des fichiers TCP sous UNIX:**

Deux fichiers principaux d'informations permettent à un processus désirant se connecter par le réseau de trouver les informations qui lui manquent pour la connexion, qu'il s'agisse d'adresses Internet distante ou de n° de port de service spécifiques. Ces fichiers se trouvent dans le répertoire /etc/ de toute machine UNIX connectée au réseau.

Le fichier /etc/hosts: contient la liste de toutes les stations du réseau, y compris la station locale. Ce fichier liste d'une part toutes les adresses Internet accessibles dans le réseau, et indique pour chacune un nom de station ( chaîne de caractères unique sur le réseau) remplaçant dans certains cas l'adresse Internet. Chaque ligne contient les informations d'une station.

Le fichier `/etc/services` : contient les n° de port (adresses de service) des services réseau disponibles sous UNIX.

## Le protocole UDP

Le protocole UDP ( User Datagram Protocol) est le deuxième protocole de transport de TCP/IP. Il assure également les communications de bout en bout mais de manière non fiable: pas d'acquittement, ni de retransmission, ni de contrôle de séquentialité, ni de contrôle de flux. D'où possibilité de perte, de duplication de message ou de rupture de séquence. Un paquet UDP est semblable au datagramme IP, il utilise les adresses Internet Source et Destination (IP) ainsi que les numéros de port Source et Destination.

UDP est recommandé pour des services nécessitant un haut niveau de performance (car du fait de sa simplicité, UDP consomme peu de temps CPU) sans besoin de fiabilité dans la transmission des données.

ANNEXE D

Éditée par L'ASSOCIATION FRANÇAISE DE NORMALISATION (AFNOR) — Tour EUROPE CEDEX 7 92080 PARIS LA DÉFENSE — Tél. (1) 778-13-28

NORME FRANÇAISE  
ENREGISTRÉE

DOCUMENTATION  
PRINCIPES GÉNÉRAUX POUR  
L'INDEXATION DES DOCUMENTS

NF  
Z 47-102  
Août 1978

SOMMAIRE

	Page
AVANT-PROPOS .....	2
1 OBJET ET DOMAINE D'APPLICATION .....	2
2 PROCESSUS ET FINALITÉ DE L'INDEXATION .....	2
3 RECONNAISSANCE DES CONCEPTS CONTENANT L'INFORMATION .....	3
3.1 APPRÉHENSION DU DOCUMENT .....	3
3.1.1 Document écrit .....	3
3.1.2 Autres documents .....	3
3.2 IDENTIFICATION DES CONCEPTS .....	4
3.3 SÉLECTION DES CONCEPTS .....	4
3.3.1 Étendue ou exhaustivité de l'indexation .....	4
3.3.2 Spécificité de l'indexation .....	5
4 REPRÉSENTATION DES CONCEPTS DANS LE LANGAGE DOCUMENTAIRE .....	5
5 CONTRÔLE DE LA QUALITÉ DE L'INDEXATION .....	6
Annexe 1 : Schéma : l'indexation dans la chaîne documentaire .....	7
Annexe 2 : Lexique .....	8

Enregistrée par décision  
du 1978-08-28  
pour prendre effet le  
1978-08-01

© AFNOR 1978  
Droits de reproduction  
et de traduction réservés  
pour tous pays

Afnor 76224

NF Z 47-102 1<sup>er</sup> Tirage 78-07

General methods for analyzing documents and determining their subjects.  
Indexierung von Dokumenten — Allgemeine Prinzipien.

## AVANT-PROPOS

La présente norme a été établie à partir des travaux sur l'indexation effectués par le Comité technique « Documentation » de l'ISO (Organisation internationale de normalisation) et l'UNESCO.

### 1 OBJET ET DOMAINE D'APPLICATION

Cette norme a pour objet d'établir des méthodologies valables et cohérentes pour la description et la caractérisation des documents à l'aide de représentations de concepts (mots-clés, descripteurs, indices, etc.).

Ces principes sont conçus de telle manière qu'ils soient dans toute la mesure du possible indépendants de tout système documentaire. Ils constituent une base uniforme de règles ou de recommandations qui doivent permettre :

- une coopération plus facile entre les différents services d'information ;
- l'établissement de règles adaptées à des systèmes particuliers.

Ces recommandations doivent permettre une indexation adaptée à un système courant de recherche d'information dans un fonds documentaire. Mais chaque système doit aller plus loin dans le raffinement par l'élaboration de règles plus spécifiques adaptées aux besoins de ses utilisateurs, tout en restant compatibles avec ces principes généraux.

### 2 PROCESSUS ET FINALITÉ DE L'INDEXATION

— L'indexation est l'opération qui consiste à décrire et à caractériser un document à l'aide de représentations des concepts contenus dans ce document, c'est-à-dire à transcrire en langage documentaire les concepts après les avoir extraits du document par une analyse. La transcription en langage documentaire se fait grâce à des outils d'indexation tels que thésaurus, classification, etc.

Le présent texte ne concerne que l'indexation du contenu intellectuel du document et ne prend pas en compte les éléments caractérisant le support physique tels que le nombre de pages, le format, le type de document...

— La finalité de l'indexation est de permettre une recherche efficace des informations contenues dans un fonds de documents et d'indiquer rapidement, sous une forme concise, la teneur d'un document.

L'indexation conduit à l'enregistrement des concepts contenus dans un document, sous une forme organisée et facilement accessible c'est-à-dire à la confection d'outils de recherche documentaire tels que des index de livres ou de bulletins bibliographiques, des catalogues organisés de matières ou encore des fichiers manuels, mécaniques ou automatiques. La recherche des informations enregistrées, par exemple la sélection de documents répondant à une question s'opère à partir de ces outils de recherche documentaire. Cette opération est possible par indexation de la question, opération analogue à l'indexation du fonds. L'indexation de la question permet d'accéder aux outils de recherche documentaire créés par l'indexation du fonds de documents.

L'utilisation d'un langage documentaire pour ces deux opérations permet d'obtenir une coïncidence exacte du vocabulaire d'indexation des documents avec celui des questions auxquelles ces documents apportent une réponse.

Un schéma donné en annexe permet de situer l'indexation (son processus et ses buts) au sein de la chaîne documentaire.

Les principes d'indexation, objet de cette norme, sont valables aussi bien dans le cas où les opérations se déroulent manuellement que dans le cas où le système est plus évolué c'est-à-dire où il y a automatisation de la recherche documentaire.

Le processus de l'indexation se déroule en deux étapes essentielles :

- la reconnaissance des concepts contenant l'information dans les documents à indexer ;
- la représentation de ces concepts dans le langage documentaire.

### 3 RECONNAISSANCE DES CONCEPTS CONTENANT L'INFORMATION

La reconnaissance des concepts contenant l'information dans un document peut elle-même se subdiviser en trois phases, sans que la limite entre ces phases soit très nette, les opérations qu'elles comportent pouvant se superposer dans le temps.

Ces trois phases sont les suivantes :

- appréhension du contenu total du document ;
- identification des concepts représentant ce contenu ;
- sélection des concepts nécessaires à la recherche ultérieure.

#### 3.1 APPRÉHENSION DU DOCUMENT

La prise de connaissance du contenu d'un document diffère selon le type de document à indexer.

##### 3.1.1 Document écrit

C'est le cas le plus fréquent. En effet, les documents écrits (livres, articles de revues, ...) forment la plus grande partie des fonds des centres de documentation et des bibliothèques.

L'appréhension du document se fait par la lecture. Mais il est important de souligner que la lecture in extenso du document n'est pas indispensable, ceci pour des raisons économiques. L'indexeur ne doit cependant perdre aucune information utile contenue dans le document et il doit repérer les passages les plus riches en information tels que :

- le titre, le résumé ;
- l'introduction, les premières phrases des chapitres et paragraphes ;
- les graphiques, les tableaux de chiffres, les figures et leurs légendes ;
- la conclusion ;
- les notions mises en valeur par certains artifices typographiques (soulignage, caractères gras, ...).

Tous ces éléments du document représentent autant de points d'ancrage que l'indexeur devra prendre en compte pour identifier l'information.

Il est recommandé d'indexer à partir du document original, le titre ou le résumé ne contenant souvent qu'un aspect ou qu'une partie ou qu'une généralisation de l'information. Dans le cas d'une indexation à partir d'un titre ou d'un résumé, le problème de l'identification du contenu informatif du document est reporté sur l'analyste qui élabore le résumé ou sur l'auteur qui doit fournir un titre synthétisant au mieux ses travaux.

##### 3.1.2 Autres documents

La démarche d'appréhension du document est différente dans le cas d'un document audiovisuel, visuel ou sonore, ou d'un objet à trois dimensions (une pièce de musée, une pièce archéologique, etc.). Les éléments à prendre en compte pour identifier l'information pourront être autres que ceux retenus dans le cas d'un document écrit.

S'il existe un document écrit, accompagnant le document ou l'objet, la démarche sera facilitée.

### 3.2 IDENTIFICATION DES CONCEPTS

Après la prise de connaissance du document, il est nécessaire que l'indexeur suive un schéma logique qui le guidera dans le choix des concepts décrivant le mieux l'information. Les principaux concepts à identifier peuvent être rangés dans un petit nombre de classes que chaque système documentaire peut préciser suivant le domaine couvert en distinguant nettement les catégories auxquelles appartiennent les concepts : s'agit-il d'un procédé ou d'un équipement ? d'une propriété ou d'un phénomène ?, etc.

L'indexeur identifie les concepts dans le document à partir d'un schéma adapté à son domaine.

Par exemple :

- pour les documents traitant de la thérapeutique médicamenteuse, l'indexeur devra rechercher le nom de la maladie, le nom du médicament, son mode d'administration, le résultat obtenu ;
- pour des documents traitant de la synthèse de composés chimiques l'indexeur devra identifier le composé de départ, le procédé de fabrication, les conditions opératoires, le produit obtenu,...

### 3.3 SÉLECTION DES CONCEPTS

Tous les concepts identifiés par l'indexeur comme représentatifs de l'information contenue dans le document, ne sont pas nécessairement retenus pour l'indexation. En effet, l'indexation doit essentiellement être adaptée à sa fonction propre dans le système où elle est utilisée. Elle peut servir pour un fonds donné de documents, à la confection d'un catalogue organisé de matières ou d'un bulletin bibliographique, à la recherche documentaire par des moyens mécaniques ou par ordinateur ou encore à la publication d'index, et elle peut alors être différente selon le type de produits qui doit en résulter.

L'adaptation de l'indexation à sa fonction et au domaine couvert par le document peut être obtenue par une variation des paramètres de l'indexation qui sont son étendue ou exhaustivité et sa spécificité.

#### 3.3.1 Étendue ou exhaustivité de l'indexation

D'une façon générale, tous les concepts contenus dans le document et pouvant être pris en compte par le système documentaire doivent se retrouver dans l'indexation. Si un document traite de plusieurs sujets indépendants mais entrant dans le domaine couvert, il peut ou bien être indexé par plusieurs spécialistes ou bien être divisé en plusieurs parties indexées séparément.

Il est recommandé de se garder d'une optique trop étroite car l'indexation d'un document peut être destinée à plusieurs centres et ce cas se multipliera avec le développement des réseaux d'information. Ainsi dans un système scientifique et technique, il ne faut pas négliger l'indexation d'aspects non techniques tels que des notions économiques ou des facteurs sociaux, ou l'indexation d'aspects marginaux.

Dans tous les cas, le critère de sélection d'un concept doit être sa signification en tant qu'élément du contenu informatif du document. Ceci revient à se demander à quelles questions le document donnera une réponse pertinente et à s'assurer que les notions appropriées figurent dans l'indexation. Ce critère est fondamental car il résulte de la finalité même de l'indexation.

Il est essentiel que l'indexation couvre les besoins des usagers. Il est utile que l'indexeur connaisse le type de questions posées au système. Il ne faut pas négliger l'effet rétroactif des questions des utilisateurs sur l'indexation au fur et à mesure qu'apparaissent de nouveaux centres d'intérêt. Il faut quand même éviter que cette adaptation aux besoins de l'utilisateur n'aille jusqu'à la déformation de l'indexation.

Théoriquement, le nombre de descripteurs par document ne doit pas être limité arbitrairement mais doit seulement dépendre de la quantité d'information contenue dans le document. Une telle limitation nuirait à l'objectivité de l'indexation et aboutirait dans beaucoup de cas à une perte d'information importante pour les utilisateurs. Par contre, pour certains systèmes documentaires cette limitation peut être dictée par des considérations économiques. Dans ce cas, le choix de l'indexeur doit être fonction de l'importance relative du contenu informatif des concepts à retenir.

Il faut également tirer du document les concepts qui y sont implicitement contenus, et qui serviront à replacer les descripteurs dans un contexte approprié.

Exemple :

« Mesure du monoxyde de carbone émis par un moteur tournant au ralenti » implique la notion de « pollution atmosphérique ».

#### 3.3.2 Spécificité de l'indexation

Elle est variable suivant les buts de l'indexation à l'intérieur d'un système documentaire. Le niveau d'indexation restera plus général pour un catalogue de bibliothèque que pour un outil de recherche documentaire destiné à une recherche rétrospective fine.

Dans le cas où l'indexation doit répondre à plusieurs objectifs, il est alors possible de couvrir tous les niveaux d'indexation du plus fin au plus général, permettant de répondre aux demandes très spécifiques ou se situant à un niveau plus général.

Il peut exister par ailleurs une relation entre le degré de spécificité de l'indexation et le degré de développement des concepts inclus par l'auteur dans un document. Une notion peu développée ou seulement citée par l'auteur si elle est retenue par l'indexeur, pourrait être indexée à un niveau de généralité élevé.

## 4 REPRÉSENTATION DES CONCEPTS DANS LE LANGAGE DOCUMENTAIRE

Il s'agit de formuler, en langage documentaire, à l'aide d'outils d'indexation, les concepts préalablement sélectionnés. Cette transcription se réalise aussi bien pour le fonds documentaire que pour la question. Elle conduit à l'organisation de l'information pour la rendre utilisable et accessible.

Il est donc recommandé de disposer d'outils d'indexation permettant de représenter l'information selon un certain nombre de règles spécifiques à chaque outil d'indexation. Les outils d'indexation les plus employés se divisent en deux groupes :

- outils de type « combinatoire » tels que thésaurus, lexique structuré, liste de sujets, etc. ;
- outils du type « catégoriel » où les concepts sont représentés par les indices ou les symboles de la classification utilisée.

Le processus d'utilisation de ces outils d'indexation se définit comme la reconnaissance dans les vocabulaires contrôlés ou les classifications, des descripteurs ou indices permettant de représenter fidèlement les concepts sélectionnés dans les documents. L'utilisation des outils d'indexation est régie par des principes propres à chaque outil.

L'indexeur devra donc avant toute indexation bien connaître l'outil dont il dispose et les contraintes particulières qui lui sont imposées ainsi que les avantages dont il dispose. Ainsi, si l'outil d'indexation est un thésaurus, le nombre de descripteurs affectés à un document peut-être réduit sans perte d'information, les relations génériques et autres pouvant être entrées directement à partir du thésaurus.

L'utilisation d'un thésaurus implique une indexation par les descripteurs les plus spécifiques.

Certains systèmes documentaires utilisent des outils du type combinatoire admettent l'utilisation d'une pondération des descripteurs, d'un chaînage entre certains descripteurs, d'indicateurs de rôle, etc. Des règles particulières précises devront être formulées dans ce cas.

Si l'indexation se fait à l'aide d'outils de type catégoriel, les indices les plus spécifiques sont toujours situés dans un contexte plus large.

L'utilisation complémentaire des deux types d'outils d'indexation peut être bénéfique et faciliter ensuite la recherche des documents. Ainsi un brevet pourra être indexé par des descripteurs mais aussi de façon complémentaire par un indice de classification.

L'indexeur sera confronté au problème de formulation de concepts nouveaux ne pouvant être traduits dans le vocabulaire contrôlé ou dans la classification. Il faudra donc prévoir une procédure de mise à jour de l'outil d'indexation selon le système utilisé.

## 5 CONTRÔLE DE LA QUALITÉ DE L'INDEXATION

La qualité de l'indexation dépend de deux facteurs :

- la qualité de l'indexeur,
- la qualité de l'outil d'indexation.

Dans le cas idéal, pour un système donné, l'indexation d'un document devrait être identique quel que soit l'indexeur. Elle ne devrait pas non plus varier dans le temps pour un même indexeur si l'outil documentaire n'a pas été modifié. Une variation dans le temps ne peut être corrigée que par une rétroaction sur le système documentaire.

Pour que l'indexation soit cohérente, l'indexeur doit posséder un certain nombre de qualités dont la principale est l'impartialité. Il doit minimiser les effets, difficilement évitables, d'un jugement subjectif. Les difficultés d'obtention d'une bonne cohérence sont d'autant plus grandes que l'équipe d'indexation est plus nombreuse d'une part, et que les indexeurs sont plus éloignés les uns des autres (cas de l'indexation décentralisée), d'autre part. Dans ce cas un contrôle centralisé doit être mis en place.

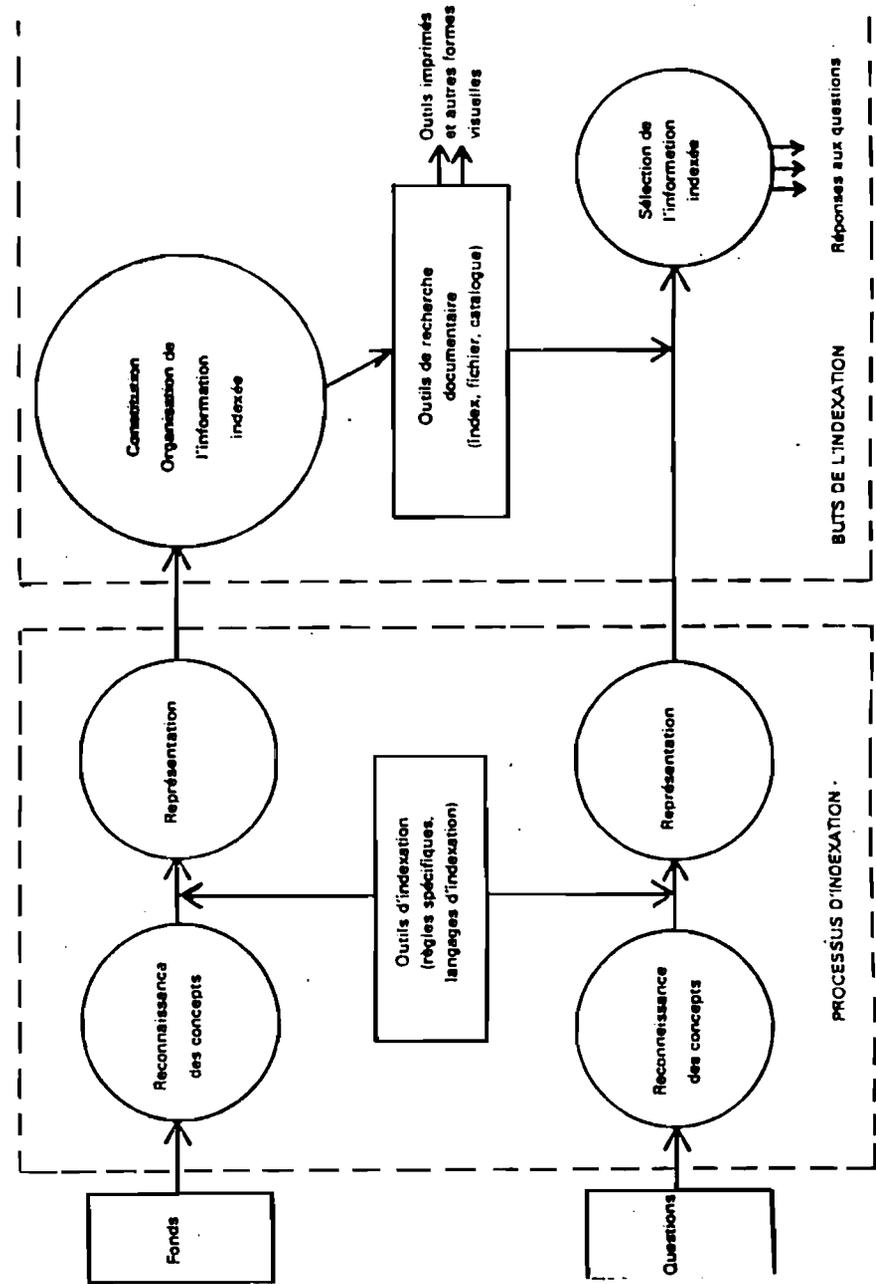
Il est nécessaire que l'indexeur soit un spécialiste de la branche dans laquelle il travaille et qu'il en connaisse le vocabulaire spécifique dans la langue des documents qu'il indexe, aussi bien que les règles et procédures spécifiques au système qu'il utilise.

La qualité de l'indexation peut être améliorée si l'indexeur connaît les types de questions posées. Il peut ainsi vérifier la qualité de son indexation en se mettant dans la situation de celui qui recherche une information. Il pourra par exemple vérifier si les descripteurs sélectionnés peuvent donner de fausses combinaisons créent du bruit à la recherche.

La qualité de l'indexation dépend aussi des qualités de l'outil documentaire utilisé. Sa qualité essentielle étant l'adaptation aux besoins nouveaux qui peuvent apparaître, et qui se traduiront par une mise à jour périodique de l'outil, dont la fréquence sera fonction du système.

Le contrôle de la qualité de l'indexation peut se faire par une analyse des résultats de recherche documentaire, par exemple à l'aide du calcul des taux de rappel et de précision.

ANNEXE 1 : L'INDEXATION DANS LA CHAÎNE DOCUMENTAIRE



LEXIQUE

Les définitions ci-après sont données à titre provisoire en attendant la publication des normes de Vocabulaire de la documentation.

CLASSIFICATION

Structuration de notions en classes et subdivisions pour expliquer les relations sémantiques généralement hiérarchiques existant entre elles au moyen d'une notation.

CONCEPT (voir : NOTION)

DESCRIPTEUR

Mot ou groupe de mots retenus dans un thésaurus et choisis parmi un ensemble de termes équivalents pour représenter sans ambiguïté une notion contenue dans un document ou dans une demande de recherche documentaire.

DOCUMENT

Ensemble d'un support d'information, des données enregistrées sur ce support et de leur signification, servant à la consultation, l'étude, la preuve, etc.

IDENTIFICATEUR

Nom propre ou assimilé utilisé comme descripteur.

**NOTE :** Ce terme peut être un nom de projet, de personne ou de collectivité, de marque de fabrique, un nom géographique, une abréviation ou un sigle, etc.

INDEX

Ensemble ordonné de termes choisis et figurant dans un document avec une indication permettant de les y localiser.

INDEXATION

Représentation par les éléments d'un langage documentaire ou naturel, des notions résultant de l'analyse du contenu d'un document en vue d'en faciliter la recherche.

MOT CLÉ

Mot choisi dans le titre ou le texte d'un document, caractérisant son contenu et permettant la recherche de ce document.

NOTION

Toute unité de pensée.

THÉSAURUS

Vocabulaire contrôlé et dynamique de termes (descripteurs et non-descripteurs) obéissant à des règles terminologiques propres et reliés entre eux par des relations sémantiques.

**NOTE :** Ce vocabulaire est employé pour représenter sans ambiguïté dans un langage artificiel des notions exprimées en langage naturel et contenues dans des documents ou dans des demandes de recherche documentaire.

3-0. 21-

# ANNEXE E : Bordereau de saisie - indexation

**HORIZON** Base bibliographique ORSTOM

1-80

NUM 77 DA 9/10/24  
aa mm jj

INV1 D 8249 : \_\_\_\_\_ INV2 \_\_\_\_\_ : \_\_\_\_\_ MIC M

NBEX1 1 NBEX2 \_\_\_\_\_ DOCEXT C; M

COTE1 D EEAD. TRE COTE2 \_\_\_\_\_

NIVBIB M A S TYPBIB (forme physique du document)

B Monographie J Article périod.  
 C Norme R Rapport officiel  
 G Carte/Atlas/Notice

INDBIB (type de document et éléments du contenu) BASES EXT

K Congrès	Z Bibliographie	I IBISCUS
L Glossaire/Encyc.	R Synth. bibliogr.	A AGRIS
Q Répertoire/Ann.	Y Cartes	AS ASFA
N Tableaux num.	O Rapport d'activité	
U Thèse, diplôme	V Multigr/Non conv.	
E Résumés		

LAN FRE : \_\_\_\_\_ : \_\_\_\_\_ : \_\_\_\_\_ : \_\_\_\_\_ ISBN \_\_\_\_\_

LANR FRE : ENG : \_\_\_\_\_ : \_\_\_\_\_ : \_\_\_\_\_

NIV S V  
 VERSION Pa. Pr.  
 CONFID 0 1  
 BULL1 1

DIFF 1 Pas d'accès possible  
 2 Consultation à l'ORSTOM  
 4 Diffusion payante auprès de l'ORSTOM  
 5 Diffusion payante sous formes mic. ou photocop.  
 6 Prêt possible

## AUTEUR

### Document générique

AUM

<u>AUM</u> /1	/2		
/3		/4	/5 _____
/1	/2		
/3		/4	/5 _____

AFM /1ORSTOM/2Institut Français de Recherche Scientifique pour le Développement en Coopération/3 /4 /5 \_\_\_\_\_

/1ORSTOM/2Institut Français de Recherche Scientifique pour le Développement en Coopération/3 /4 /5 \_\_\_\_\_

**Document générique**

AJA /Trico, Bancud

AJA /1 /2  
/3  
/1 /2  
/3

/4 /5

/4 Bamako /5 — ML

AJA /1ORSTOM/2Institut Français de Recherche Scientifique pour le Développement en Coopération/3 PPA /4 Daka /5 SEA :  
/1ORSTOM/2Institut Français de Recherche Scientifique pour le Développement en Coopération/3 /4 /5

**TITRE**

**Document générique**

TITORM

NOEDI

FRAM

TITANM

**Document spécifique**

TITORA Les Dégâts d'oiseaux d'eau sur les rigoles aménagées du delta central du Niger au Mali

FRAA

TITANA

**CONGRES**

NUMCG

NUMCG

LOCG

PAYSCG

DACG

\_\_\_\_ / \_\_\_\_ / \_\_\_\_  
8888 88 JJ

**DIPLOME**

DIPL

UNIV

DADIPL

\_\_\_\_ / \_\_\_\_ / \_\_\_\_  
8888 88 JJ

NODIPL

**OPERAT**

OPERAT

**NOBRA**

NOBRA

**COLLATION**

ILL BIBLIOGR CARTE \_\_\_\_:1/\_\_\_\_:\_\_\_\_:1/\_\_\_\_:\_\_\_\_:1/\_\_\_\_:\_\_\_\_:1/\_\_\_\_  
TABL GRAPH PHOTO MATDAC DICT

LED (\_\_\_\_) ED  
DP \_\_\_\_ / \_\_\_\_ / \_\_\_\_ VOLO \_\_\_\_ / \_\_\_\_ PAGE

**SERIE**

JT Journal d'Agriculture Traditionnelle FRA PAGE p. 153-170  
COLL et de Botanique Appliquée (\_\_\_\_)  
VOL ~~34~~ NUMER \_\_\_\_ ISSN \_\_\_\_ DP 1987/\_\_\_\_

NOTE

GRAUPLAØ3

**DESCRIPTEURS ORSTON**

CHAP ~~(8) ZOOGENO3~~ \_\_\_\_

DES  
oiseau ornemental ; seizeuse ; de gats ;  
oiseau nuisible ; partie de récolte ; méthode  
CDE de lutte ; protection des rizières ; Zone humide

GEO Mali

LOC Niger delta Central ; Niger Cours d'eau

## LISTES SYSTEMATIQUES

Indiquer le nom du champ: *quelca quelca*

AN1: *Anas querquedula*  
*Philomachus pugnax*

---

## DESCRIPTEURS BASSES EXTERIEURES

DAGRIS

CHAPAGR

DIBIS

---

## RESUME

Cette étude porte sur les dégâts d'oiseau d'eau dans le delta central du Niger, et sur la détermination du niveau des dégâts, de leurs modalités et des espèces responsables. Des moyens de protéger les rizières et de diminuer les pertes sont proposés.

```

#!/bin/sh

#
#=====#
#
# infoserveur_horizon
# Recupere l'adresse de l'expediteur
# Active verif_syntaxe pour le controle des requetes
# Active traitement_requete pour l'interrogation
#
#=====#
#

#
# VARIABLES
#
# OPTIONS
NomMailServeur="horizon@dakar.orstom.sn"
Repertoire="/usr2/user/fianyoy/version1"
Rep_tmp="/usr2/user/fianyoy/tempo"
Gestionnaire="fianyoy"
# Fin des OPTIONS
#
MAIL="/usr/lib/sendmail -f horizon"
Adresse=""
# Fin des VARIABLES
#
# FICHIERS
TmpRequete="$Rep_tmp/requete$$"
MessRepcnse="MessReponse$$"
MessBad="$Rep_tmp/MessBad$$"
MauvForm="$Rep_tmp/MauvForm$$"
MessAvert="$Rep_tmp/MessAvert$$"
MessErreur="$Rep_tmp/MessErr$$"
Reponse="$Rep_tmp/Reponse$$"
Bad="$Rep_tmp/Bad$$"
Good="$Rep_tmp/Good$$"
Format="$Rep_tmp/Format$$"
# Fin des FICHIERS

cat - > $TmpRequete

#
# Recherche de l'adresse de l'expediteur
#
RechExpediteur()
{
    #
    # VARIABLES GLOBALES
    # TmpRequete
    # Fin des VARIABLES GLOBALES
    #
    # PARAMETRES
    #
    # Adresse
    # Fin des PARAMETRES
    #
    # VARIABLES LOCALES
    #

```

```

Expediteur=""
ExpediteurNom=""
ExpediteurAdresse=""
# Fin des VARIABLES LOCALES

#
# Recherche de la ligne expéditeur dans le mail selon la norme
#
Expediteur=""grep \Sender: \ $TmpRequete | cut -d\ " -f2-10"
if [ "$Expediteur" = "" ]
then
    Expediteur=""grep \Reply-To: \ $TmpRequete | cut -d\ " -f2-10"
    if [ "$Expediteur" = "" ]
    then
        Expediteur=""grep \From: \ $TmpRequete | cut -d\ " -f2-10"
        if [ "$Expediteur" = "" ]
        then
            Expediteur=""grep \From \ $TmpRequete | cut -d\ " -f2-10"
            fi
        fi
    fi
fi

#
# Recherche de l'adresse de l'expéditeur.
#
if [ "$Expediteur" != "" ]
then
    # Les adresses INTERNET completes
    if [ ""echo $Expediteur | grep \@"" != "" ]
    then
        ExpediteurNom=""echo $Expediteur | cut -d'@' -f1"
        while [ ""echo $ExpediteurNom | grep '"" != "" ]
        do
            ExpediteurNom=""echo $ExpediteurNom | cut -d' ' -f2-10"
        done
        ExpediteurNom=""echo $ExpediteurNom | cut -d'<' -f2"
        if [ "$ExpediteurNom" = "" ]
        then
            $MAIL -s "Err infoserveur : $Expediteur" $Gestionnaire < $TmpRequete
            exit 1
        fi

        ExpediteurAdresse=""echo $Expediteur | cut -d'@' -f2"
        while [ ""echo $ExpediteurAdresse | grep '"" != "" ]
        do
            ExpediteurAdresse=""echo $ExpediteurAdresse | cut -d' ' -f1"
        done
        ExpediteurAdresse=""echo $ExpediteurAdresse | cut -d'>' -f1"
        if [ "$ExpediteurAdresse" = "" ]
        then
            $MAIL -s "Err infoserveur : Expediteur" $Gestionnaire < $TmpRequete
            exit 1
        fi
    fi
else
    # Les adresses incompletes
    Adresse=""echo $Expediteur | cut -d' ' -f1"
fi

```

```

fi
}

#
# Composition du message de reponse
#
ComposeMessRep()
{
    #
    # VARIABLES GLOBALES
    # TmpRequete, MessReponse, Reponse
    # Fin des VARIABLES GLOBALES
    #

    echo "MAIL-SERVEUR \"\$NomMailServeur\"" > $MessReponse
    echo " >> $MessReponse
    echo '----- DEBUT DE VOTRE REQUETE -----' >> $MessReponse
    cat $Good >> $MessReponse
    echo '----- FIN DE VOTRE REQUETE -----' >> $MessReponse
    echo " >> $MessReponse
    echo " >> $MessReponse
    echo '----- DEBUT REPONSE -----' >> $MessReponse
    cat $Reponse >> $MessReponse
    echo '----- FIN REPONSE -----' >> $MessReponse
}

```

```

#
# Composition du message de mauvaise(s) requete(s)
#
ComposeMessBad()
{
    #
    # VARIABLES GLOBALES
    # MessBad, Bad
    # Fin des VARIABLES GLOBALES
    #

    echo "MAIL-SERVEUR \"\$NomMailServeur\"" > $MessBad
    echo " >> $MessBad
    echo '----- LISTE DE(S) REQUETE(S) SYNTAXIQUEMENT INCORRECTE(S)-----'>>
$MessBad
    cat $Bad >> $MessBad
    echo '-----' >> $MessBad
}

```

```

#
# Composition du message d'avertissement en cas de compression du fichier
#
ComposeMessAvert()
{
    #
    # VARIABLES GLOBALES
    # MessAvert
    # Fin des VARIABLES GLOBALES

    echo "MAIL-SERVEUR \"\$NomMailServeur\"" > $MessAvert
}

```

```

echo "                                >> $MessAvert
echo '-----' >> $MessAvert
echo "                                >> $MessAvert
echo ' VOUS ALLEZ RECEVOIR UN MESSAGE CODE' >> $MessAvert
echo ' POUR LE RENDRE LISIBLE, SAUVEZ CE MESSAGE DANS UN FICHER'
>> $MessAvert
echo "                                >> $MessAvert
echo ' PUIS (SOUS UNIX) FAITES decodage <nom fichier>' >> $MessAvert
echo "                                >> $MessAvert
echo '-----' >> $MessAvert

```

}

```

#
# Composition du message en cas de mauvais format du fichier de requetes

```

```

#
ComposeMauvFormat()

```

{

```

#
# VARIABLES GLOBALES
# MauvForm, Format
# Fin des VARIABLES GLOBALES
#

```

}

```

echo "MAIL-SERVEUR \"${NomMailServeur}\" > $MauvForm
echo "                                >> $MauvForm
echo '-----' >> $MauvForm
cat $Format                                >> $MauvForm
echo

```

'-----' >>

```

$MauvForm
echo "                                >> $MauvForm
cat $TmpRequete                                >> $MauvForm
echo '-----' >> $MauvForm

```

}

```

#
# Avertissement en cas d'erreur systeme

```

```

#
ErreurSysteme()

```

{

```

#
# VARIABLES GLOBALES
# MessErreur
# Fin des VARIABLES GLOBALES
#
# PARAMETRES
# FICHER Repertoire nom-programme
# Fin des PARAMETRES

```

```

echo "MAIL-SERVEUR \"${NomMailServeur}\" > $MessErreur
echo "                                >> $MessErreur
echo "ERREUR SYSTEME pendant execution " >> $MessErreur
echo "      Votre demande ne peut etre satisfaite" >> $MessErreur
echo "                                >> $MessErreur

```

```

echo ' VOTRE FICHER : '>> $MessErreur
cat $1                                >> $MessErreur
echo "                                >> $MessErreur
echo '-----' >> $MessErreur
$MAIL $Adresse < $MessErreur
echo "                                >> $MessErreur
echo "      Fichier retourne a : $Adresse" >> $MessErreur
echo "      ERREUR : $2/$3" >> $MessErreur
$MAIL $Gestionnaire < $MessErreur
echo "                                >> $MessErreur
echo '-----' >> $MessErreur
/bin/rm $1 MessErreur

```

```

}

```

```

#
# PROGRAMME PRINCIPAL
# =====
#

```

```

RechExpediteur

```

```

if [ "$Adresse" != "" ]
then

```

```

    if [ "$Adresse" != "$NomMailServeur" ]
    then

```

```

        $Repertoire/verif_syntaxe $TmpRequete $Bad $Good $Format $Rep_tmp
        case $? in

```

```

            1) /bin/rm $Bad $Good $Format
               ErreurSysteme $TmpRequete $Repertoire verif_syntaxe
               exit -1 ;;

```

```

            3) ComposeMauvFormat
               $MAIL $Adresse < $MauvForm
               /bin/rm $MauvForm $Format
               exit 3 ;;

```

```

        esac

```

```

        if test -f $Bad

```

```

        then

```

```

            ComposeMessBad
            $MAIL $Adresse < $MessBad
            /bin/rm $Bad $MessBad

```

```

        fi

```

```

        if test -f $Good

```

```

        then

```

```

            $Repertoire/traitement_requete $Good $Reponse $Rep_tmp
            case $? in

```

```

                3|0) ComposeMessRep
                   $MAIL $Adresse < $MessReponse;;

```

```

                4) ComposeMessRep
                   ComposeMessAvert
                   $MAIL $Adresse < $MessAvert
                   ---> codage $MessReponse $Adresse
                   /bin/rm $MessAvert;;

```

```

                *) /bin/rm $Reponse $TmpRequete
                   ErreurSysteme $Good $Repertoire traitement_requete
                   exit -1 ;;

```

```
        esac
    fi
else
    $MAIL $Gestionnaire < $TmpRequete
    exit 1
fi
/bin/rm $TmpRequete $Good $MessReponse
else
    $MAIL $Gestionnaire < $TmpRequete
    /bin/rm $TmpRequete
    exit 1
fi
exit 0
```

Yawa Edem FIANYO
------------------

loginfos2 : programme LOGOTEL d'interrogation de la base horisen
--

```

module .A
action .utiliser pquestio
    .question
    .utiliser ficquest
    .question
    .utiliser ficpar
    .question
    .utiliser basedoc
    .question
    .utiliser icrep
    .question
    .utiliser edform
    .question
    .utiliser ednotice
    .question
    .utiliser nbremax
    .question
    .utiliser option
    .question
    .utiliser ficnbrerep
    .question
    .utiliser ficcat
    .question
    .modifier nbre : 100
    .modifier reptot : 1000
    .modifier fin_prema : 0
    .modifier var : [somme [nbremax] [reptot]]
    .aller INITIAL
comment .chargement du nom des fichiers de travail (fichier question, fichier
    .reponse, fichier documentaire a consulter avec son fichier parametre,
    .fichiers de mise en forme des resultats avec leurs parametres) et
    .initialisation des variables de travail (numero de la question a
    .charger, nombre de reponse deja trouvee)

```

```

module .CHARGQU
action .si reptot =>[var]
    .aller FINPREMA
    .etiquette CH_QUESTION
    .modifier nbre : [somme [nbre] 1]
    .modifier nbreb :[apres 1 [nbre]]
    .texto
    .texto /[pquestio]
    .texto /f [ficquest]
    .texto qv
    .texto NUM=[nbreb]
    .si texto=.
    .aller ZFIN
    .si fin_prema=1
    .aller FINPREMA ,IMPMESS
    .etiquette CH_CHARG
    .modifier qcour :[LETTRE]
    .texto
    .texto
    .si tin_prema=2

```

```

    .aller FINPREMA ,IMPQUES
    .aller VERNBREC
comment .lecture dans le document virtuel (ficquest) de la question a traiter.
    .Si il n'y en a plus, on va a ZFIN.Le 2eme texto sert a sortir du
    .document virtuel, le 3eme a sortir de ques. Le 1er texto sert a
    .sortir du ques de REPINF10

module .FINPREMA
action .imprimer-fichier * [ficrep]
    .imprimer-fichier
    .
    .LE FICHER DE REPONSE CONTIENT [somme [nbremax] 0] REFERENCES.
    .AUCUNE AUTRE REQUETE NE PEUT PLUS ETRE TRAITEE.
    .
    .imprimer-fichier *
    .modifier fin_prema : [somme [fin_prema] 1]
    .aller CHARGQU ,CH_QUESTION
    .etiquette IMPMESS
    .imprimer-fichier * [ficrep]
    .imprimer-fichier
    .REQUETE(S) NON TRAITEE(S) :
    .
    .imprimer-fichier *
    .modifier fin_prema : [somme [fin_prema] 1]
    .aller CHARGQU ,CH_CHARG
    .etiquette IMPQUES
    .imprimer-fichier * [ficrep]
    .imprimer-fichier
    . -> [qcour]
    .imprimer-fichier *
    .aller CHARGQU ,CH_QUESTION
comment .lorsqu'il y a deja nbremax (=100) references dans le fichier de
    .reponse, on ne traite plus les requetes du fichier question mais on
    .ajoute dans le fichier reponse les requetes non traitees si elles
    .existent.

```

```

module .INITIAL
action .texto mode bref
    .texto cat [ficcat]
    .aller CHARGQU
comment .initialisation de l'environnement de travail de texto.

```

```

module .MAUVREQ
action .imprimer-fichier * [ficrep]
    .imprimer-fichier
    .VOTRE REQUETE =====> [qcour]
    .
    . TEXTO NE COMPREND PAS CETTE REQUETE,
    . ELLE EST PEUT ETRE AMBIGUE
    .
    . ***** FIN DE LA REQUETE *****
    . *****
    .imprimer-fichier *
    .aller CHARGQU
comment .cas des requetes incorrectes ou ambiguës

```

```

module .PASDEREP
action .imprimer-fichier * [ficrep]
    .imprimer-fichier

```

```

.VOTRE REQUETE =====> [qcour]
.
.AUCUNE REFERENCE NE CORRESPOND A VOTRE REQUETE
.
.***** FIN DE LA REQUETE *****
.*****
.imprimer-fichier *
.aller CHARGQU
comment .traitement du cas de requete ne donnant aucune reference.

```

```

module .PLUSDE10
action .imprimer-fichier * [ficrep]
.imprimer-fichier
.VOTRE REQUETE =====> [qcour]
.
.PLUS DE 100 REFERENCES CORRESPONDENT A CETTE REQUETE,
.ELLE NE PEUT ETRE TRAITEE.
.
.***** FIN DE LA REQUETE *****
.*****
.imprimer-fichier *
.aller CHARGQU
comment .traitement de requete qui donne plus de 100 references

```

```

module .REPINF10
action .imprimer-fichier * [ficrep]
.imprimer-fichier
.VOTRE REQUETE =====> [qcour]
.
.[somme [reponse] 0] REFERENCE(S) CORRESPONDANTE(S):
.imprimer-fichier *
.modifier option : [somme [option 0]]
.si option=1
.texto /[ednotice]
.aller ,FINIE
.texto /[edform]
.etiquette FINIE
.texto q2re
.texto [ficrep]
.texto [qcour]
.texto
.imprimer-fichier * [ficrep]
.imprimer-fichier
.***** FIN DE LA REQUETE ( [qcour] ) *****
.imprimer-fichier *
.aller CHARGQU
comment .si le nombre de reference est dans les limites permises, on ajoute au
.fichier reponse ces references.

```

```

module .REPSUP10
action .imprimer-fichier * [ficrep]
.imprimer-fichier
.VOTRE REQUETE =====> [qcour]
.NOMBRE DE REFERENCES =====> [somme [reponse] 0]
.
.LENREGISTREMENT DES REFERENCES CORRESPONDANT A VOTRE REQUETE
.ENTRAINERAIT UN DEPASSEMENT DE LIMITE DU FICHER DE REPONSE.
.RELANCER L'INFOSERVEUR POUR LE TRAITEMENT DE CETTE REQUETE
.

```

```

.***** FIN DE LA REQUETE *****
.*****
.imprimer-fichier *
.aller CHARGQU
comment .Si le nombre de reponses total depasse 100, on le signale dans le
.fichier de reponse sans envoyer les references correspondantes

module .VERNBREC
action .texto /{ficpar}
.texto /f [basedoc]
.texto init
.texto ques
.texto [qcour]
.texto
.modifier txt : [somme {texto$} 0]
.si txt=0
.aller MAUVREQ
.si reponse0=00000000
.aller PASDEREP
.modifier var2 : [scmme 100000000 [nbremax]]
.modifier var2 : [apres 1 [var2]]
.si reponse0=>[var2]
.aller PLUSDE10
.aller VNBRETOT
comment .on charge la base documentaire et on lui soumet la requete. On teste
.le nombre de reponse obtenu. la commande texto sert a sortir de
.l'environnement ques.

module .VNBRETOT
action .modifier reptmp : [somme [reptot] [reponse]]
.si reptmp=>[somme [var] 1]
.aller REPSUP10
.modifier reptot : [reptmp]
.aller REPINF10
comment .Test sur le nombre de reponse qui figureront dans le fichier de
.reponse

module .ZFIN
action .imprimer-fichier * [ficrep]
.imprimer-fichier
.***** FIN DU TRAITEMENT DES REQUETES
.*****
.imprimer-fichier *
.imprimer-fichier * [ficnbrerep]
.imprimer-fichier
.[somme [reptot] -1000]
.imprimer-fichier *
.texto init
.texto fin
.
comment .fin du programme. On recupere le nbre total de reponses dans un
.fichier en enlevant 1000 a reptot car on a initialise reptot avec
.cette somme

```

Les programmes en langage C sont présentés sous la forme de fichiers annotés obtenus par `tcov`. Ces fichiers permettent d'obtenir la couverture du code

De tels fichiers montrent le nombre d'appel de chaque fonction, (`####` signifie que la fonction n'a jamais été appelée). A la fin du programme se trouve un résumé sur le nombre de blocs exécutés.

Les programmes `traitement_requete` et `verif_syntaxe` ont été exécuté avec en entrée un fichier contenant un peu plus de 100 requêtes.

```
/*
```

Yawa Edem FIANYO - Juin 1993 - ULIS Dakar
exam_BAL.c
programme d'examen de la BAL horizon (info-serveur horizon)
ENTREES : 1. ARGUMENT : nom du programme de traitement des messages (infoserveur_horizon) 2. FICHIER : - fichier /var/spool/mail/horizon : fichier BAL de horizon
SORTIES : 1. CODES : 0 si aucun message a traiter, 1 si de nouveaux messages ont ete traites
DESCRIPTION : La boite aux lettres horizon est scrutee. Les nouveaux messages sont transmis a infoserveur_horizon

```
*/
```

```
# include <stdio.h>
# include <string.h>
# define NBMAX_NMSG 50
```

```
char fmailbox[80],
     fcmd[80],
     nom_exe[20],
     nompgm[80];
int nb_mes=0,
    commande=0,
    tmessage[NBMAX_NMSG];
```

```
#include "fonctions.h"
```

```
/*-----*/
```

```
nouveau_message(chaine)
char *chaine;
36 -> { int trouve=0, i=0;

       while ( (!trouve) && (i<3) && (chaine[i] !='N') && (chaine[i] !='U') )
2 ->   { i++; }
36 ->   if (i < 3) trouve = 1;
36 ->   return (trouve);
}
```

```
/*-----*/
```

```
enregistrer_message(num)
int num;
36 -> {
     tmessage[nb_mes]=num;
     nb_mes++;
     return;
}
```

```

/*-----*/

usage(nom)
char *nom;
##### -> { printf("\n%s: USAGE \n%s <nompgm> \n\n",nom_exe, nom_exe);
    exit(-1);
}

/*-----*/

recuperer_BAL()
2 -> { char cmd[80];
    FILE *fp;

    if ( (fp=fopen(fcmd,"w"))==NULL ) sortie(nom_exe,fcmd);
2 -> fprintf(fp,"q\n");
    fclose(fp);
    sprintf(cmd,"mail > %s < %s",fmailbox,fcmd);
    system(cmd);
    sprintf(cmd,"/bin/rm %s",fcmd);
    system(cmd);
    return;
}

/*-----*/

traiter_message()
1 -> { int i;
    FILE *fp;
    char cmd[80];

    if ( (fp=fopen(fcmd,"w"))==NULL ) sortie(nom_exe,fcmd);
1 -> for (i=0; i<nb_mes; i++)
18 -> { fprintf(fp,"%d %s\n", tmessage[i], nompgm); }
1 -> fprintf(fp,"q\n");
    fclose(fp);
    sprintf(cmd,"mail < %s",fcmd);
    system(cmd);
    sprintf(cmd,"/bin/rm %s",fcmd);
    system(cmd);
    return;
}

/*-----*/

effacer_fmailbox()
2 -> { char cmd[80];

    sprintf(cmd,"/bin/rm %s",fcmd);
    system(cmd);
    sprintf(cmd,"/bin/rm %s",fmailbox);
    system(cmd);
}

/*-----*/

main(argc, argv)
int argc;
char *argv[];

```

```

2 -> { FILE *fp;
      char buf[257],
          etat[3],
          expediteur[80];
      int numsg, pid;

      pid=getpid();
      strcpy(nom_exe,argv[0]);
      if (argc !=2) usage(nom_exe);
2 ->   strcpy(nompgm,argv[1]);
      sprintf(fmailbox,"BAL%d",pid);
      sprintf(fcnd,"CDE%d",pid);
      recuperer_BAL();
      if ( (fp=fopen(fmailbox,"r"))==NULL ) sortie(nom_exe,fmailbox);
2 ->   while ( fgets(buf,257,fp) != NULL )
46 ->     { prep(buf);
          if ( sscanf(buf,"%3s %d %s %*s %*s %*d %*s %*s",
                      etat, &numsg ,expediteur)==3 )
36 ->         { if ( nouveau_message(etat) ) enregistrer_message(numsg) ; }

          }
2 ->   effacer_fmailbox();
      if (nb_mes)
2 ->     {if (fork()==0)
1 ->         { traiter_message();
              exit(1);
            }
1 ->     else exit(1);
      }
##### -> exit(0);
      }

```

#### Top 10 Blocks

Line	Count
138	46
46	36
50	36
51	36
58	36
141	36
97	18
49	2
75	2
79	2

23 Basic blocks in this file  
 21 Basic blocks executed  
 91.30 Percent of the file executed  
  
 267 Total basic block executions  
 11.61 Average executions per basic block



```

/*-----*/

verification(buf)
char *buf;
1 -> {FILE *fp;

    if (verif_fichier(buf))
1 -> { if ( (fp = fopen(fgood,"a"))== NULL ) sortie(nom_exe,fgood);
1 ->    fprintf (fp,"requete horizon\n");
        fclose(fp);
        return (1);
    }
##### -> return (0);
}

/*-----*/

creer_fic_temp()
1 -> { int pid = getpid();
    sprintf (fic_inter,"%s/temp%d",rep_tmp,pid);
    sprintf (fic_entree,"%s/entree%d",rep_tmp,pid);
    return;
}

/*-----*/

effacer_fic_temp()
1 -> {char cmd[256];
    sprintf (cmd,"/bin/rm %s %s", fic_inter, fic_entree);
    system(cmd);
    return;
}

/*-----*/

verif_ligne_requete(buf)
char *buf;
108 -> {FILE *fm, *fentree;

    if (option_aide_fichier(buf))
##### -> { if ( (fm = fopen(fgood,"a"))== NULL ) sortie(nom_exe,fgood);
##### ->    fprintf (fm, "aide fichier\n");
        fclose (fm);
        return (0);
    };
108 -> if (option_aide_question(buf))
##### -> { if ( (fm = fopen(fgood,"a"))== NULL ) sortie(nom_exe,fgood);
##### ->    fprintf (fm, "aide question\n");
        fclose (fm);
        return (0);
    };
108 -> if (option_format(buf))
##### -> { if ( (fm = fopen(fgood,"a"))==NULL ) sortie(nom_exe,fgood);
##### ->    fprintf (fm, "format brut\n");
        fclose (fm);
        return (0);
    };
}

```

```

108 -> if (option_fin(buf))
1 -> { if ( (fm = fopen(fgood,"a"))== NULL ) sortie(nom_exe,fgood);
1 ->   fprintf (fm, "fin\n");
      fclose (fm);
      arret = 1;
      return (0);
    };
107 -> if (option_compresse(buf))
##### -> { if ( (fm = fopen(fgood,"a"))== NULL ) sortie(nom_exe,fgood);
##### ->   fprintf (fm, "compresse\n");
      fclose (fm);
      return (0);
    };
107 -> if ( (fentree = fopen(fic_entree,"w"))==NULL) sortie(nom_exe,fic_entree);
107 -> fprintf(fentree,"%s\n",buf);
      fclose(fentree);
      return (1);
    }

/*-----*/

# include "y.tab.c"

/*-----*/

copie_fic(fic_in,fic_out)
char *fic_in,*fic_out;
107 -> { FILE *fin, *fout;
      int c;
      fin = fopen(fic_in, "r");
      fout= fopen(fic_out,"a");
      while ( (c=getc(fin)) != EOF )
3514 ->   putc (c,fout);
      /* if ( c != '\n' ) putc ('\n',fout);*/
107 ->   fclose(fout);
      fclose(fin);
      return;
    }
/*-----*/

mauvais_format()
##### -> {FILE *fm;
      if ( (fm=fopen(fmformat,"w"))==NULL ) sortie(nom_exe,fmformat);
##### ->   fprintf(fm,"\nVOTRE FICHER DE REQUETE N'EST PAS AU BON FORMAT.\n");
      fprintf(fm,"IL DOIT COMMENCER PAR requete horizon ET SE TERMINER PAR fin\n\n");
      fclose(fm);
      printf("\nLe fichier mail est incorrect\n");
      exit (3);
    }
/*-----*/

main (argc,argv)
int argc;
char *argv[];
1 -> {
      char buf[257];
      int Ok=0;

      strcpy(nom_exe, argv[0]);

```

```

if ( argc != 6 ) usage(nom_exe);
1 -> strcpy(fmail, argv[1]);
strcpy(fbad, argv[2]);
strcpy(fgood, argv[3]);
strcpy(fmformat, argv[4]);
strcpy(rep_tmp, argv[5]);
if ( (verif_format(fmail))!=2 ) mauvais_format();
1 -> if ( (fe=fopen(fmail,"r"))==NULL ) sortie(nom_exe,fmail);
1 -> while ( !Ok )
1 ->     { if ( fgets(buf,257,fe) != NULL )
1 ->         { prep(buf);
           Ok = verification(buf);
           }
           };
1 -> creer_fic_temp();
while ( (arret != 1) && (fgets(buf,257,fe) != NULL) )
108 ->     { if ( (prep(buf) > 2) && (verif_ligne_requete(buf)) )
107 ->         { if ( yyparse (fic_entree, fic_inter) )
9 ->             copie_fic(fic_inter,fbad);
98 ->         else copie_fic(fic_inter,fgood);
           };
           }
1 -> eifacer_fic_temp();
exit(0);
}

```

Top 10 Blocks

Line	Count
153	3514
101	108
109	108
115	108
121	108
196	108
128	107
134	107
135	107
148	107

41 Basic blocks in this file  
29 Basic blocks executed  
70.73 Percent of the file executed

4818 Total basic block executions  
117.51 Average executions per basic block

Yawa Edem FIANYO - ULIS - avril 1993

Spécification de la grammaire (version provisoire)

```
%{
# include <stdio.h>
# include <ctype.h>
%}

%start requete

%token CHAMPS MOT TRANSEXP TRANSREQ

%%
requete      : requetecomp requetesimple
             | requetesimple
             ;

requetesimple : '(' requeteprim ')'
              | requeteprim
              ;

expressionsimple : '(' expression ')'
                 | phrase ;

phrase          : phrase MOT
                 | MOT
                 ;

requeteprim : CHAMPS '=' expression
             | CHAMPS '=' expressionsimple
             ;

expression : expressioncomp expressionsimple
           ;

expressioncomp : expressionsimple TRANSEXP ;

requetecomp : requete TRANSREQ
            | '(' requete ')' TRANSREQ;

%%
# include "lex.yy.c"
```

Yawa Edem FIANYO - ULIS - mai 1993

Spécification de l'analyseur lexical (version provisoire)

```

CHAINE      [a-zA-Z0-9<> \.!'i-_\]*
CHAINE1     [a-zA-Z0-9<> \.!'i-_\]{CHAINE}
ESPACE      [\t]
DES         ([Dd][Ee][Ss][Mm][Oo][Tt]){CHAINE}{ESPACE}*
DP          [Dd]([Pp][Aa][Tt][Ee]){CHAINE}{ESPACE}*
AU          [Aa][Uu]{CHAINE}{ESPACE}*
GEO         ([Gg][Ee][Oo][Ll][Oo][Cc][Ll][Ii][Ee][Uu]){CHAINE}{ESPACE}*
CHAP        [Cc][Hh][Aa][Pp]{CHAINE}{ESPACE}*
ET          [Ee][Tt]{ESPACE}*
OU          [Oo][Uu]{ESPACE}*
SAUF        [Ss][Aa][Uu][Ff]{ESPACE}*
MOTCLE      {CHAINE}{ESPACE}* \=
    
```

%a 3000

%o 4000

%%

```

{DES}/"=" {fprintf(yyval,"DES"); return CHAMPS;}
{DP}/"=" {fprintf(yyval,"DP"); return CHAMPS;}
{AU}/"=" {fprintf(yyval,"AU"); return CHAMPS;}
{GEO}/"=" {fprintf(yyval,"GEO"); return CHAMPS;}
{CHAP}/"=" {fprintf(yyval,"CHAP"); return CHAMPS;}
"="{ESPACE}* {fprintf(yyval,"="); return '=';}
{SAUF}/"(""{MOTCLE} {fprintf(yyval," SAUF "); return TRANSREQ;}
{OU}/"(""{MOTCLE} {fprintf(yyval," OU "); return TRANSREQ;}
{ET}/"(""{MOTCLE} {fprintf(yyval," ET "); return TRANSREQ;}
{SAUF} {fprintf(yyval," SAUF "); return TRANSEXP;}
{OU} {fprintf(yyval," OU "); return TRANSEXP;}
{ET} {fprintf(yyval," ET "); return TRANSEXP;}
{CHAINE1}{ESPACE}* {int i;
    for (i=0; i<yyleng; i++) fprintf
        (yyval,"%c",toupper(yytext[i]));
    return MOT;
}
"(""{ESPACE}* {fprintf(yyval,"("); return '(';}
")""{ESPACE}* {fprintf(yyval,")"); return ')'}
""["^"]*"" {ECHO; return MOT; }
    
```

Yawa Edem FIANYO - Juin 1993 - ULIS Dakar

traitement\_requete.c

programme de traitement des requetes correctes (info-serveur horizon)

**ENTREES :**

1. ARGUMENTS: a) nom du fichier ou se trouvent les requetes a traiter  
b) nom du fichier ou doivent etre deposees les reponses  
c) nom du repertoire ou sont crees les fichiers de travail
2. FICHIERS : - fichier de requetes correctes  
- fichier documentaire TEXTO a interroger  
- fichiers d'aide
3. AUTRES : - nom des documents parametres, documents d'edition necessaires a TEXTO pour travailler  
- nom du programme LOGOTEL d'interrogation  
- nom des fichiers d'aide  
- repertoire de travail de TEXTO ou se trouve le fichier paratext contenant les documents d'edition et de parametres necessaires au bon deroulement du programme LOGOTEL

**SORTIES :**

1. CODES : 0 si Ok,  
4 si Ok et compression du fichier de reponse demandee,  
3 si mauvais format du fichier de requetes  
1 si nombre d'arguments incorrects ou erreur systeme : probleme d'ouverture d'un fichier  
terminaison anormale de TEXTO
2. FICHIERS : -le fichier de reponses

**DESCRIPTION :**

Le fichier de requetes correctes est decortique pour separer les options des requetes a soumettre. Celles-ci sont deposees dans un fichier ASCII puis transferees par une commande TEXTO dans un fichier exploitable par le programme LOGOTEL. Si un fichier d'aide est demande, il est concatene au fichier de reponses. Si le nombre de questions a soumettre est non nul, on active le programme LOGOTEL d'interrogation. Il a pour resultat de creer un fichier de reponses de TEXTO qui est concatene au fichier de reponses.

```
#include <stdio.h>
#include <string.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/param.h>
```

```
char nom_exe[8],
      doc_courant[20],
      fmail[80],
      rep_tmp[80],
      freponse[80],
      frequete[80],
      fnbrerep[80];
```

```
FILE *fe;
int format=0,
    aide_quest=0,
    aide_fic=0,
    fin=0,
    compresse=0,
    commande=0,
    tic_req=0,
    nbremax=100,
```

```

    nbrequest=0,
    pid,
    nbrerep;

# include "horizon.h"
# include "fonctions.h"
/*-----*/

void usage(nom_exe)
char *nom_exe;
##### -> {
    printf("\n%s: USAGE: \n%s <fichier-mail> <fic-reponse> <rep_travail> \n\n",nom_exe,nom_exe);
    exit(1);
}
/*-----*/

ajout_fic_requete(buf)
char *buf;
196 -> { FILE *fr;

    if (!fic_req) { sprintf (frequete,"%s/req%d",rep_tmp,pid);
2 ->         fic_req = 1;
                }
196 -> if ( (fr=fopen(frequete,"a")==NULL )
##### ->     sortie(nom_exe,frequete);
196 -> if ( strlen(buf)>2 ) { fprintf(fr,"%s\n",buf);
196 ->         nbrequest++;
                }
196 -> fclose(fr);
    return;
}
/*-----*/

void aide_format()
##### -> {
    strcpy(doc_courant,"AIDE FICHIER REQUETE:");
    envoi(fic_aide1);
    exit(3);
}
/*-----*/

enleve_accents(fichier)
char *fichier;
##### -> {
    char      cmd[256],
              fcmd[20],
              fic[20];
    FILE *pf;

    sprintf(fcmd,"%s/cdes%d",rep_tmp,pid);
    sprintf (fic,"%s/trav%d",rep_tmp,pid);
    pf=fopen(fcmd,"w");
    fprintf (pf, "tr "β""_L—lø Ø caaaaeeeeiiou < %s > %s\n", fichier, fic);
    fprintf (pf, "/bin/rm %s\n",fichier);
    fprintf (pf, "mv %s %s\n",fic, fichier);
}

```

```

fclose(pf);
sprintf(cmd,"sh %s",fcmd);
system (cmd);
sprintf(cmd, "/bin/rm %s", fcmd);
system (cmd);
return;
}

```

/\*-----\*/

```

envoi(fichier)
char *fichier;

```

```

1 -> { char cmd[256];
      FILE *pf, *pg;

      if ( (pf=fopen(freponse,"a"))==NULL ) sortie(nom_exe,freponse);
1 ->  if ( (pg=fopen(fichier,"r"))!=NULL )
1 ->    { fclose(pg);
        /* enleve_accents(fichier);*/
        fprintf(pf,"\n===> %s <===\n\n",doc_courant);
      }
1 ->  fclose(pf);
      sprintf(cmd,"cat %s >> %s",
              fichier,
              freponse);
      system(cmd);
      return;
}

```

/\*-----\*/

```

envoi_aide1()
##### -> {
  strcpy(doc_courant,"AIDE FICHER REQUETE");
  envoi(fic_aide1);
  return;
}

```

/\*-----\*/

```

envoi_aide2()
##### -> {
  strcpy(doc_courant,"AIDE FORMULATION DES QUESTIONS");
  envoi(fic_aide2);
  return;
}

```

/\*-----\*/

```

traitement_texto()
1 -> {
  FILE *fc;
  char      cmd[256],
            fcmd[80],
            ficquest[80],
            frep[80];

  sprintf(fcmd,"%s/cmd%d",rep_tmp,pid);

```

```

    sprintf(ficquest,"%s/fich%d",rep_tmp,pid);
    sprintf(frep,"%s/frep%d",rep_tmp,pid);
    sprintf(fnbrerep,"%s/nbre%d",rep_tmp,pid);
    if ( (fc=fopen(fcmd,"w")) == NULL )
##### ->   sortie(nom_exe,fcmd);
1 ->   fprintf(fc,"dial yyy\n");
        fprintf(fc,"cat %s\n",ficcat);
        fprintf(fc,"%s\n",ficpques);
        fprintf(fc,"%f %s\n",ficquest);
        fprintf(fc,"a\n");
        fclose(fc);
        sprintf(cmd,"cat %s >> %s",frequete,fcmd);
        system(cmd);
        fc=fopen(fcmd,"a");
        fprintf(fc,"ctr\n");
        fprintf(fc,"%f %s\n",prglog);
        fprintf(fc,"%s\n",ficpques);
        fprintf(fc,"%s\n",ficquest);
        fprintf(fc,"%s\n",ficpar);
        fprintf(fc,"%s\n",basedoc);
        fprintf(fc,"%s\n",frep);
        fprintf(fc,"%s\n",fedit1);
        fprintf(fc,"%s\n",fedit2);
        fprintf(fc,"%d\n",nbremax);
        fprintf(fc,"%d\n",format);
        fprintf(fc,"%s\n",fnbrerep);
        fprintf(fc,"%s\n",ficcat);
        fclose(fc);
        sprintf (cmd,"%s < %s",commande_texto,fcmd);
        system (cmd);
        strcpy(doc_courant,"REPONSES DE TEXTO");
        envoi(frep);
        sprintf (cmd,"/bin/rm %s",frep);
        system (cmd);
        sprintf (cmd,"/bin/rm %s",ficquest);
        system (cmd);
        sprintf (cmd,"/bin/rm %s",fcmd);
        system (cmd);
        sprintf (cmd,"/bin/rm %s",frequete);
        system (cmd);
        if ( (fc=fopen(fnbrerep,"r"))==NULL ) sortie(nom_exe,fnbrerep);
1 ->   fgets(cmd,80,fc);
        fgets(cmd,80,fc);
        fclose(fc);
        sscanf(cmd,"%d",&nbrerep);
        sprintf(cmd,"/bin/rm %s",fnbrerep);
        system(cmd);
        return;
    }

/*-----*/

pas_de_quest()
##### -> { FILE *fp;
        if ( (fp=fopen(freponse,"a"))==NULL ) sortie (nom_exe,freponse);
:##### ->   fprintf(fp,"\n\n N'Y A AUCUNE QUESTION A SOUMETTRE A TEXTO\n");
        fclose(fp);
        if (!aide_quest) envoi_aide2();
##### ->   exit(0);

```

```

}
/*-----*/
main (argc,argv)
int argc;
char *argv[];
2 -> {
    char buf[257];

    int Ok = 0;
    int termine=0,
        p[2];

    pid=getpid();
    strcpy(nom_exe,argv[0]);
    if ( argc != 4 ) usage(nom_exe);
2 ->   strcpy(fmail,argv[1]);
    strcpy(freponse,argv[2]);
    strcpy(rep_tmp,argv[3]);
    if ( (verif_format(fmail))!=2 ) aide_format();
2 ->   if ( (fe=fopen(fmail,"r"))==NULL ) sortie(nom_exe,fmail);
2 ->   while ( !Ok )
2 ->       { if ( (fgets(buf,257,fe))!=NULL )
2 ->           {      prep(buf);
                Ok = verf_fichier(buf);
            }
        }
2 ->   while ( (!fin) && (fgets(buf,257,fe))!=NULL )
198 ->   {      prep(buf);
        format += option_format(buf);
        if ( !commande ) aide_quest += option_aide_question(buf);
198 ->   if ( !commande ) aide_fic += option_aide_fichier(buf);
198 ->   if ( !commande ) fin += option_fin(buf);
198 ->   if ( !commande ) compresse += option_compresse(buf);
198 ->   if ( !commande ) ajout_fic_requete(buf);
198 ->   commande = 0;
    }
2 ->   if ( aide_fic ) envoi_aide1();
2 ->   if ( aide_quest ) envoi_aide2();
2 ->   if ( format ) format = 1;
2 ->   fclose(fe);
    if (!nbrequest) pas_de_quest();
2 ->   if ( pipe(p) ) sortie(nom_exe, "pipe");
2 ->   if ( (fork()) > 0 )
    { struct rusage *temps;
      1 ->   getrusage(RUSAGE_SELF,temps);
        close(p[1]);
        while (!termine)
            { if ( ((*temps).ru_utime.tv_sec) > 310 ) break;
              1 ->   termine=read(p[0],buf,8);
                    getrusage(RUSAGE_SELF,temps);
            };
        1 ->   if ( !termine ) exit(1);
        1 ->   if ( compresse || (nbrerep > 50) )      exit(4);
##### ->   exit(0);
            }
            else
            {
                1 ->   close(p[0]);
            }
    }
}

```

```

    traitement_texto();
    write(p[1],"termine",8);
    close(p[1]);
    exit(0);
}

```

##### -> }

Top 10 Blocks

Line	Count
274	198
277	198
278	198
279	198
280	198
281	198
89	196
94	196
96	196
97	196

49 Basic blocks in this file  
 37 Basic blocks executed  
 75.51 Percent of the file executed  
  
 2208 Total basic block executions  
 45.06 Average executions per basic block

Nom : FIANYO

Prénoms : Yawa Edem

Titre : Etude et réalisation d'un Info-serveur sur la base HORIZON

Dakar, 1993

## RESUME

Ce mémoire porte sur l'étude et la réalisation d'une application informatique permettant à tout utilisateur du Réseau Informatique de l'ORSTOM (R.I.O.) de consulter de façon différée une base documentaire ORSTOM (HORIZON).

Les différents modes d'accès au R.I.O. et l'exploitation d'une base bibliographique sont présentés.

Les phases de la réalisation du logiciel sont détaillées selon la méthode classique: A partir d'un Cahier des Charges qui présente le problème à résoudre et les principales caractéristiques de la solution à mettre en place, les Spécifications dégagent les outils matériels et logiciels qui sont utilisés ainsi que les grands blocs qui forment l'application. Les parties Conception et Interface présentent les caractéristiques techniques des fonctions implémentées : algorithme, données en entrées et en sortie, exception à traiter, taille. La partie Tests décrit les performances du logiciel et présente les résultats des différents tests qui ont été réalisés.

Les manuels d'utilisation et d'exploitation sont aussi présentés.

**Mots-clés** : informatique documentaire; info-serveur; HORIZON;  
réseau informatique; recherche documentaire; recherche  
de l'information; logiciel.