



**HAL**  
open science

## Conférence répartie en mode messagerie

Philippe Brun

► **To cite this version:**

Philippe Brun. Conférence répartie en mode messagerie. Web. Ecole Nationale Supérieure des Mines de Saint-Etienne; Université Jean Monnet - Saint-Etienne, 1997. Français. NNT : 1987STET4010 . tel-00941202

**HAL Id: tel-00941202**

**<https://theses.hal.science/tel-00941202>**

Submitted on 3 Feb 2014

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

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

ECOLE NATIONALE SUPERIEURE  
DES MINES DE SAINT-ETIENNE

N° d'ordre : 15-ID

PL 9/40 11

## THESE

présentée par

**Philippe BRUN**

pour obtenir le titre de

DOCTEUR

DE L'UNIVERSITE DE SAINT-ETIENNE

ET DE L'ECOLE DES MINES DE SAINT-ETIENNE

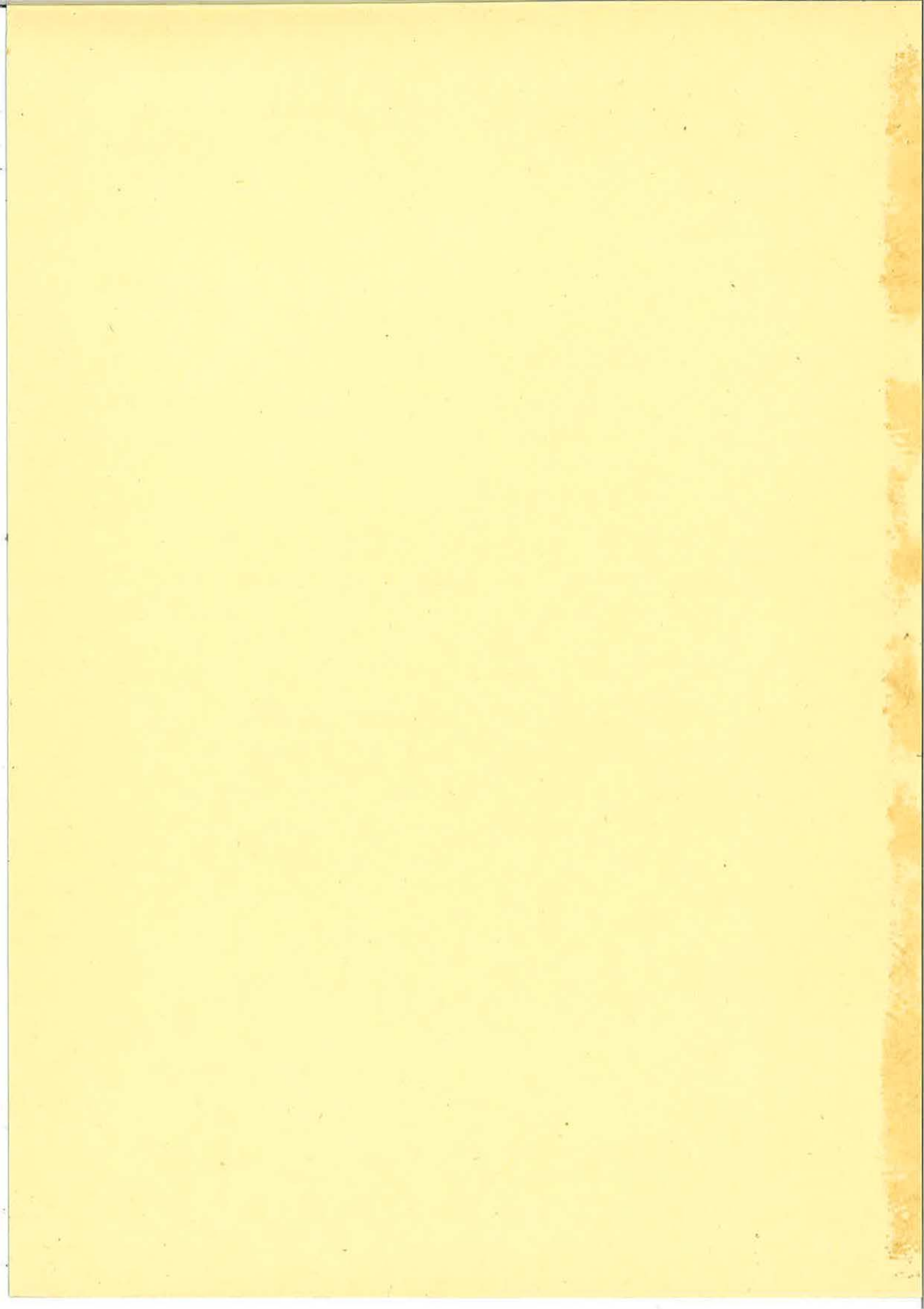
Spécialité : Informatique, Image, Intelligence Artificielle et Algorithmique

## CONFERENCE REPARTIE EN MODE MESSAGERIE

Soutenue à Saint-Etienne, le 9 novembre 1987 devant le jury :

Président G. PUJOLLE

Examineurs J. AZEMA  
C. KINTZIG  
P.A. PAYS  
B. PEROCHE  
P. VAN BINST



**THESE**

présentée par

**Philippe BRUN**

pour obtenir le titre de

**DOCTEUR**

**DE L'UNIVERSITE DE SAINT-ETIENNE**

**ET DE L'ECOLE DES MINES DE SAINT-ETIENNE**

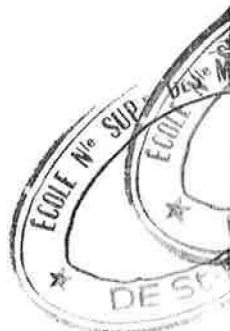
Spécialité : Informatique, Image, Intelligence Artificielle et Algorithmique

**CONFERENCE REPARTIE  
EN MODE MESSAGERIE**

Soutenue à Saint-Etienne, le 9 novembre 1987 devant le jury :

Président G. PUJOLLE

Examineurs J. AZEMA  
C. KINTZIG  
P.A. PAYS  
B. PEROCHE  
P. VAN BINST





## ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT ETIENNE

Directeur : M. Philippe SAINT RAYMOND  
Directeur des Etudes et de la formation : M. Jean CHEVALIER  
Directeur des Recherches : M. MUDRY  
Secrétaire Général : M. J. Claude PIATEK

### PROFESSEURS DE 1ère CATEGORIE

MM. COINDE	Alexandre	Gestion
FORMERY	Philippe	Mathématiques Appliquées
GOUX	Claude	Métallurgie
LE COZE	Jean	Matériaux
LOWYS	Jean-Pierre	Physique
MATHON	Albert	Gestion
PERRIN	Michel	Géologie
PEROCHE	Bernard	Informatique
RIEU	Jean	Mécanique - Résistance des Matériaux
SOUSTELLE	Michel	Chimie
VERCHERY	Georges	Matériaux



### PROFESSEURS DE 2ème CATEGORIE

MM. LADET	Pierre	Entreprise et Travaux
PLA	Jean Marie	Mathématiques
TOUCHARD	Bernard	Physique Industrielle

### DIRECTEUR DE RECHERCHE

M. LESBATS	Pierre	Métallurgie
------------	--------	-------------

### MAITRES DE RECHERCHE

MM. BISCONDI	Michel	Métallurgie
CONRAD	Francis	Informatique
DAVUINE	Philippe	Géologie
DRIVER	Julian	Matériaux
Mlle FOURDEUX	Angeline	Métallurgie
MM. GIRARDOT	Jean Jacques	Informatique
GUILHOT	Bernard	Chimie
KOBYLANSKI	André	Métallurgie
LALAUZE	René	Chimie
LANCELOT	Francis	Chimie
MONTHEILLET	Franck	Matériaux
TRAN MINH	Canh	Chimie

### PERSONNALITES HABILITEES A DIRIGER LES TRAVAUX DE RECHERCHE

MM. COURNIL	Michel	Chimie
MAGNIN	Thierry	Matériaux
THOMAS	Gérard	Chimie

### PROFESSEUR A L'U.E.R. DE SCIENCES DE SAINT ETIENNE

M. VERGNAUD	Jean Marie	Chimie des Matériaux
-------------	------------	----------------------





Je tiens à exprimer ma reconnaissance à toutes les personnes qui m'ont aidé au cours de ces années de thèse.

- Monsieur le Professeur Guy Pujolle de l'Université Pierre et Marie Curie qui me fait l'honneur de présider le jury de cette thèse.
- Monsieur le Professeur Paul Van Binst de l'Université Libre de Bruxelles pour l'intérêt qu'il accorde à mes travaux.
- Monsieur Jean Azéma, Maître de Conférence à l'Université de Sciences de Saint-Etienne pour avoir accepté de participer au jury.
- Monsieur Claude Kintzig, Ingénieur au Centre National d'Etudes des Télécommunications qui est à l'origine de ce projet et qui en fut l'un des principaux animateurs.
- Monsieur Bernard Peroche, Directeur du Département Informatique de l'Ecole des Mines de Saint-Etienne pour l'intérêt porté à cette thèse.
- Monsieur Paul-André Pays, Ingénieur à l'Ecole des Mines de Saint-Etienne, pour sa compétence, sa disponibilité et la formation que j'ai reçue au cours de ces années de thèse.
- Tous mes collègues du groupe de travail qui au fil des réunions ont contribué à l'élaboration de CRMM.
- Tous les membres du département informatique pour l'amitié qu'ils ont su faire régner.
- Madame Geneviève Lallich pour l'aide apportée pendant ces années de thèse.
- Monsieur Jean Marc Piégay de la société Sync pour ses conseils dans le maniement du formatteur de texte GROFF employé pour éditer ce document.
- Le personnel du service de reprographie pour avoir assuré la réalisation de cet ouvrage.
- Mes proches, parents et amis, pour leur affection et leur soutien au cours de ces années de thèse et de celles qui les ont précédées.





## SOMMAIRE

---

### INTRODUCTION

---

### 1 : PRESENTATION DES CONFERENCES INFORMATIQUES

<b>LES CONFERENCES INFORMATIQUES</b>	<b>1</b>
Historique	1
Caractérisation d'un système de conférences	1
<b>ETAT DE L'ART</b>	<b>4</b>
Un exemple : COM	4
Centralisation du produit	5
Gestion des utilisateurs	5
Relations entre messages	6
Administration du système	6
Fonctionnalités minimales d'un système	6
<b>PUBLIC CONCERNE</b>	<b>7</b>
<b>VERS UNE PLUS LARGE UTILISATION</b>	<b>7</b>
Applications possibles	7
De nouvelles contraintes	7
Contrôle du flux des informations	7
Adaptation des conférences	8
Réduire les coûts	8
Pour un accès simplifié.	8
<b>CONCLUSION DE CETTE VUE D'ENSEMBLE</b>	<b>9</b>

---

## **2 : LE SERVICE CRMM**

<b>GESTION DE L'INFORMATION</b>	<b>12</b>
Groupes de discussion	12
Relations entre messages	12
Limiter les contraintes	13
Propositions actuelles	14
Gestion des nouveautés	14
Relations entre messages	15
Les travaux en cours	15
<b>TRAVAUX DE GROUPE ASSISTES PAR ORDINATEUR.</b>	<b>16</b>
Influences sur les organisations	16
Concepts de base	17
Le modèle AMIGO	18
Les limites du modèle	19
<b>LES PROPOSITIONS DE CRMM</b>	<b>20</b>
La base des messages	20
Les fiches de CRMM	20
Relations entre fiches	21
Opérations sur les fiches	29
Vision personnalisée de la base	34
Remarques sur l'archivage	34
CRMM exemple de communication de groupe	34
Description de CRMM	35

---

## **3 : REPARTITION DE LA CONFERENCE**

<b>EXEMPLES DE CONFERENCES DISTRIBUEES</b>	<b>42</b>
Expérience Cosac	42
TOOL	43
<b>LA SOLUTION CRMM</b>	<b>45</b>
Modèle avant Répartition	45
Liste de diffusion	46
Répartition du répertoire	46
Répartition du rôle d'administrateur	46
Duplication de la base des messages	47
Synchronisation des messages	48
Désignation	51
Désignation des fiches	51
La désignation des usagers.	52
Format des fiches	53
Réception de messages erronés	53
Les messages entre sites	54
Diffusion d'opérations sur la base	54
Message de services	54
Imprimatur	55
Messages d'erreur	55

<b>INTEGRATION DANS LA MESSAGERIE X400</b>	<b>56</b>
La messagerie	56
Fonctions d'un système de messagerie	58
La messagerie X400	59
Description générale	59
Représentation du modèle MHS	63
Syntaxe X409	64
La messagerie COSAC	65
CRMM et la messagerie	66
Conférences et courrier interpersonnel	67
Description des entités	69
Description en couche	70
Désignation	70
Le protocole	72
Primitives de BBGA offertes aux BBUA	75
Primitives de BBUA offertes à l'utilisateur	76
Quelques scénarii	77
Consultation de la base	77
Ajout d'une fiche par un écrivain	77
Ajout d'une fiche par un usager non écrivain	79
Effacement d'une fiche	81

---

## **4 : INTERFACE UTILISATEUR**

<b>LES DIFFICULTES DE L'INTERFACE</b>	<b>83</b>
Etude récente	83
Complexité des logiciels	84
Grande diffusion des produits	84
Indépendance vis-à-vis du système	85
Accès au produit	86
Personnalisation de l'interface	86
Monde de l'utilisateur et monde informatique	87
Absence d'outils pour les concepteurs	87
<b>LES NOUVELLES INTERFACES UTILISATEUR</b>	<b>88</b>
WYSIWYG	88
Les objets	89
Les menus	90
La métaphore du bureau	91
Le matériel et logiciels nécessaires	91
Spécification de l'interface	92
Boîtes à outils	93
Unité logique	93
Terminal virtuel	94
Entrées-Sortie de haut niveau	96
Les intégrateurs	97
<b>LE BUREAU CRMM</b>	<b>98</b>
Services à offrir	98
Station de travail	99
UA éclaté	100

Les objets internes de l'interface CRMM	101
La liste des numéros locaux des fiches non lues,	101
La liste des fiches précédentes	102
La liste des fiches sélectionnées	102
La liste des fiches marquées	103
La liste des alias des sites	103
Le contenu de la boîte	103
La description de la conférence	105
La description de l'ensemble des conférences	105
Les fiches disponibles localement	105
Les objets d'affichage	106
La fenêtre de lecture	106
La fenêtre de sélection de fiches	107
La fenêtre de résultat	108
La fenêtre des marques	109
La fenêtre de la boîte aux lettres	109
La fenêtre de copie	110
La fenêtre d'état	110
La fenêtre de création de fiche	111

---

## 5 : REALISATION

<b>LES OUTILS UTILISES</b>	<b>113</b>
Windows	113
Le toolkit de développement	114
GDI	115
Windows	115
Les fenêtres de Windows.	116
Fonctionnement de l'intégrateur	117
Evénements et messages	117
Fonction principale	118
Fonction de traitement des messages	118
Fenêtres prédéfinies	122
Boîtes de dialogue	122
Boîte de messages	124
Les menus	125
Les ressources	125
Réalisation d'une application windows	125
Gestion du multifenêtrage	126
<b>PRESENTATION GENERALE DES MODULES DE CRMM</b>	<b>131</b>
<b>ECLATEMENT DU MODULE IU</b>	<b>132</b>
Communication client-serveur	132
Les objets IU serveur	133
Les objets IU client	133
Les fenêtres	135
Les menus	137

---

**6 : CRMM++**

<b>NOUVELLES FONCTIONNALITES</b>	<b>142</b>
<b>IMPLANTATION</b>	<b>145</b>
<b>NOUVEAUX SERVICES</b>	<b>145</b>
<b>GENERATEUR DE CONFERENCE</b>	<b>148</b>

---

**BIBLIOGRAPHIE**

---

**ANNEXE**

---



## Introduction

### 1 CADRE DU PROJET

La thèse décrite dans ce rapport a été entreprise dans le cadre d'une collaboration du CNET (Centre National d'Etudes des Télécommunications) et de l'Ecole des Mines de Saint-Etienne (ENSMSE). Cette étude fait partie du projet SMARTIX [HOUERY85].

SMARTIX a été lancé en 1985. Il résulte de la volonté du CNET de développer une informatique répartie normalisée. Il implique comme autres partenaires Bull, l'INRIA et le CNRS. Il repose sur trois points particuliers :

- implantation rapide d'un réseau opérationnel sur le site du CNET d'Issy-les-Moulineaux,
- apport des compétences du CNET en matière de communication,
- cohérence des réseaux informatiques avec les services télématiques et les réseaux de télécommunication.

Ce projet doit conduire à des réalisations commercialisables pour les gammes de machines SM90 (devenues SPS 7 chez Bull) et Multics. Il étudie notamment :

- un réseau de base avec ses coupleurs X-25, les services de transport et session,
- le traitement de documents,



- les postes de travail,
- l'environnement réparti,
- la gestion de réseau,
- et la messagerie qui reprend le système COSAC [KINTZIG84] et ses services de courrier électronique, de transfert de fichiers, de soumission de travaux et de conférences réparties. C'est sur ce dernier point qu'intervient l'étude décrite dans cette thèse.

## 2 BUT DU PROJET

CRMM (Conférence Répartie en Mode Messagerie) s'intéresse aux conférences informatiques. Il s'agit d'outils de communication comparables à un panneau d'affichage où l'on dépose des messages pour qu'ils soient lus par la communauté des utilisateurs. Le projet tire profit de l'expérience de produits existants tels que Forum sur Multics, les *news* d'Unix ou Com à Stockholm. L'étude comprend une phase de spécification et une autre de réalisation.

Les spécifications ont été menées à bien par un groupe de travail. Outre l'Ecole des Mines et le CNET qui furent les principaux intervenants, se trouvaient représentés divers partenaires comme l'université Paris VI et la société Bull. Cette première phase a abouti en juin 1986 à la production d'un document de référence [BRUN86].

Les spécifications tirent parti de la normalisation des produits de communication, cela se traduit par le respect des propositions du CCITT (Comité Consultatif International Télégraphique et Téléphonique), plus particulièrement des recommandations X400 sur la messagerie. Cette démarche conduit à décrire CRMM comme une application normalisable. Le modèle ainsi obtenu est défini par ses éléments de service, le protocole entre les entités communicantes et les primitives qu'elles offrent.

L'étude tient compte des problèmes inhérents aux outils de communication lorsque le nombre d'utilisateurs et le volume d'informations échangées deviennent trop importants. Leur résolution suppose une gestion efficace des messages échangés et des interlocuteurs.

Le projet profite des progrès en cours dans le domaine de la bureautique. Il en découle une architecture de type client-serveur dans laquelle l'utilisateur accède à un service à travers un poste de travail. Le dialogue entre l'utilisateur et sa station de travail est amélioré grâce aux nouveaux produits disponibles actuellement : écran graphique, système de fenêtrage et outils de pointage.

Pendant cette phase, ma tâche fut d'étudier et de proposer des choix de spécifications qui ont été débattus lors des réunions mensuelles du groupe de travail. Je tenais à jour et éditais le document de référence issu de ces discussions.

Une fois le modèle CRMM établi, la phase de réalisation a débuté. Elle fut confiée à Paris VI, au CNET et à l'Ecole des Mines. J'ai pris en charge ce qui concerne le dialogue avec l'utilisateur. Pour cela, je me suis intéressé aux techniques, matériels et idées en cours dans le domaine de l'interface utilisateur. Les conclusions auxquelles j'ai abouti, ont été elles-aussi soumises au groupe de travail.

### 3 PLAN DU RAPPORT DE THESE

La première année de la thèse fut consacrée au travail de spécifications. Toutefois ce document ne les décrit ni en détail, ni de manière formelle. En effet, le document de référence [BRUN86] les reprend de façon précise et complète. En revanche, les choix sont expliqués et justifiés. De plus, cet ouvrage résume les notions nécessaires à la compréhension de la messagerie normalisée. La bibliographie donne les références pour une connaissance complète de ce mode de communication. Enfin, comme la réalisation ne porte que sur une partie du projet, les autres composants du produit ne sont pas entièrement décrits. Il seront repris en détail par deux autres thèses à venir [ZENG88] et [DJEBALI88].

Au début de cet ouvrage est donnée une présentation générale des conférences informatiques. Un exemple de système existant permet d'introduire les éléments qui les caractérisent. Ensuite, les options retenues sont expliquées et justifiées. Les choix fondamentaux portent sur la structure des messages, la gestion des utilisateurs et la répartition des conférences.

Le second chapitre présente les services offerts par CRMM pour gérer les informations et les membres d'une conférence. Il insiste sur les relations entre messages et sur les opérations qu'ils supportent. Il montre l'intérêt d'une base structurée pour les archiver. D'autre part, les conférences informatiques sont des outils destinés à des groupes d'individus. Elles influent sur l'organisation de ces groupes, notamment en imposant des règles d'utilisation. Ces aspects sont étudiés pour permettre une bonne collaboration entre les membres.

Le chapitre suivant s'intéresse aux effets de la répartition de la conférence. Il met en évidence l'intérêt d'une telle démarche et les dispositions qu'elle nécessite, tant pour la gestion des messages que pour celle des usagers. La partie suivante du chapitre traite du mode de communication utilisé. Celui qui est choisi, la messagerie X400, est décrit succinctement et le modèle de communication obtenu pour CRMM est spécifié selon les habitudes en vigueur pour ce type de produits. A l'issue de ce chapitre, le modèle du produit de messagerie CRMM est complet.

Puis, le dialogue entre l'utilisateur du produit et le système est étudié. Après un rappel des problèmes rencontrés dans ce type de dialogue et des nouvelles possibilités matérielles et logicielles, l'architecture choisie est présentée.

Ensuite, un chapitre est consacré à la réalisation des logiciels, avec une description de leurs modules et des objets manipulés. Cette section contient également une présentation rapide de l'intégrateur Windows de Microsoft qui est utilisé pour réaliser l'interface.

Enfin, la dernière partie indique les évolutions possibles et souhaitables de CRMM.

**PRESENTATION DES CONFERENCES INFORMATIQUES 1**

---

**LES CONFERENCES INFORMATIQUES 1**  
Historique 1  
Caractérisation d'un système de conférences 1

**ETAT DE L'ART 4**  
Un exemple : COM 4  
Centralisation du produit 5  
Gestion des utilisateurs 5  
Relations entre messages 6  
Administration du système 6  
Fonctionnalités minimales d'un système 6

**PUBLIC CONCERNE 7**

**VERS UNE PLUS LARGE UTILISATION 7**  
Applications possibles 7  
De nouvelles contraintes 7  
    Contrôle du flux des informations 7  
    Adaptation des conférences 8  
    Réduire les coûts 8  
    Pour un accès simplifié. 8

**CONCLUSION DE CETTE VUE D'ENSEMBLE 9**



## Chapitre 1

### PRESENTATION DES CONFERENCES INFORMATIQUES

#### 1 LES CONFERENCES INFORMATIQUES

##### 1.1 HISTORIQUE

En Automne 1971 aux Etats-Unis, le gouvernement du président Nixon réforma profondément le contrôle des prix. Ce bouleversement de la vie économique nécessita la mise en place d'un système de collecte et de diffusion d'informations à l'échelle d'un pays. Sous la conduite de Murray Turroff, EMISARI (*Emergency Management Information System and Reference Index*) vit le jour. Ce système reliait une dizaine de centres régionaux. Il permettait la consultation d'informations telles que règlements, indications, ou autres éclaircissements. Ces données pouvaient être rapidement mises à jour par les autorités fédérales. Elles étaient archivées, ainsi les lecteurs n'avaient pas à être connectés sans cesse pour ne pas perdre de messages.

Les utilisateurs pouvaient également interroger les experts de Washington qui leur répondaient toujours grâce à EMISARI. Les questions et les réponses demeuraient accessibles à l'ensemble des usagers.

##### 1.2 CARACTERISATION D'UN SYSTEME DE CONFERENCES

Les communications basées sur des systèmes informatiques sont soit synchrones, tous les interlocuteurs doivent être présents au même moment, soit asynchrones, les utilisateurs envoient et consultent les messages au moment qu'ils jugent opportun. Les vidéoconférences sont un bon exemple du premier mode. Le courrier électronique et les conférences informatiques illustrent le second.

Le courrier électronique est un outil de communication de personne à personne, alors que les conférences telles que le système EMISARI offrent à un groupe d'individus le moyen de communiquer. Contrairement au courrier qui permet un échange d'informations entre un expéditeur et un ou plusieurs destinataires, les informations transmises par EMISARI concernent un ensemble d'individus. Même lorsqu'ils répondaient à une question posée par un correspondant donné, les représentants du gouvernement s'adressaient en fait à l'ensemble des utilisateurs. Certes, le courrier permet la mult destination des messages, un ersatz de conférence est de ce fait possible : chaque correspondant dispose de la liste de tous les membres de la conférence et leur adresse tous les messages qu'il crée. Mais le courrier est mal adapté à ce mode de communication. Par exemple, lorsqu'un usager s'intègre au groupe, il doit obtenir l'historique des messages et la liste de diffusion auprès d'un utilisateur plus ancien. Il faut aussi qu'il signale son arrivée à tous les membres du groupe pour qu'ils mettent à jour leur copie de la liste des destinataires. De plus, l'organisation des messages stockés est propre à chaque usager, ce qui empêche toute référence commune. Ces remarques montrent l'intérêt d'un outil collectif.

Un tel système doit comprendre un service d'archivage pour stocker les informations échangées. Ceci permet les échanges asynchrones entre utilisateurs. Ce service est plus ou moins important selon les systèmes, mais certaines possibilités sont généralement disponibles. Par exemple, des aides comme la signalisation des textes non encore lus, sont fréquemment offertes. De plus, comme cela est décrit précédemment, les messages peuvent se répondre. Ces relations entre les textes créent des "conversations", qui seront suivies en différé par les futurs lecteurs de la conférence. Le gestionnaire de l'ensemble des messages doit prendre en compte ces liens pour faciliter leur parcours.

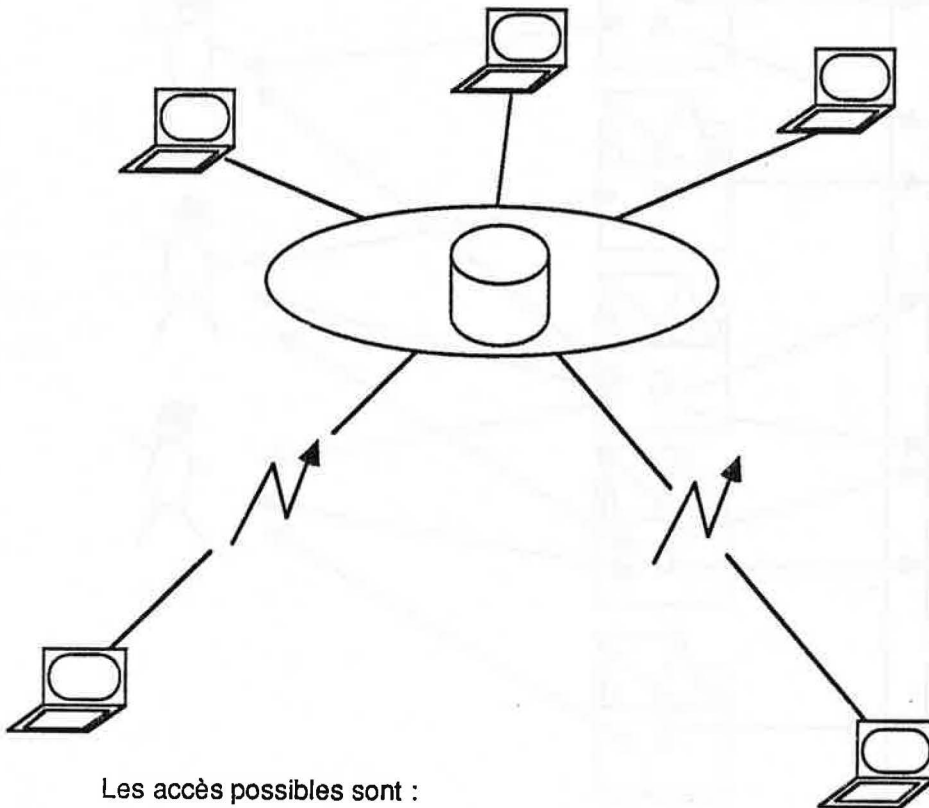
Il est évident que tous les utilisateurs n'ont pas tous des droits identiques. Par exemple, il ne sont pas tous autorisés à écrire dans certaines conférences, ou encore, à accéder à certains messages confidentiels.

Les conférences sont bien adaptées à une diffusion de l'information. En revanche, contrairement au courrier de personne à personne, il est difficile d'assurer à un auteur que ses messages seront ou ont été effectivement lus par tous les membres. Dans un système fortement centralisé, cette fonctionnalité est réalisable, mais devient lourde à gérer dès que les membres sont répartis sur différents sites.

Les conférences telles qu'elles sont étudiées dans ce recueil, reprennent les fonctionnalités d'un outil traditionnel : le panneau d'affichage sur lequel les usagers accrochent des fiches dont le contenu sera lu par les éventuels consultants. Dans la terminologie anglo-saxonne, les conférences informatiques concernent aussi les conversations en temps réel à plusieurs intervenants connectés. Le terme pour les

conférences décrites ici est *Bulletin Board*, c'est-à-dire panneau d'affichage. Comme dans le domaine de la messagerie les entités sont souvent connues sous leur désignation anglaise, le lecteur doit s'attendre à rencontrer des références à cette expression.

Pour résumer ce rapide tour d'horizon, nous retenons les caractéristiques suivantes. Les conférences informatiques permettent à un groupe d'individus d'échanger des informations sur un sujet commun. Le système de communication est bâti sur une base de messages qui est lue ou modifiée par un groupe d'utilisateurs. Les accès se font à partir d'un terminal quelconque relié par réseau local ou à grande distance (Fig. I.1). Les produits se différencient par leur gestion de la base, notamment les outils qu'ils offrent aux utilisateurs pour créer, modifier, ajouter, consulter les messages, et par les options disponibles dans l'attribution des droits aux usagers.



Les accès possibles sont :

- locaux,
- distants à partir d'un terminal

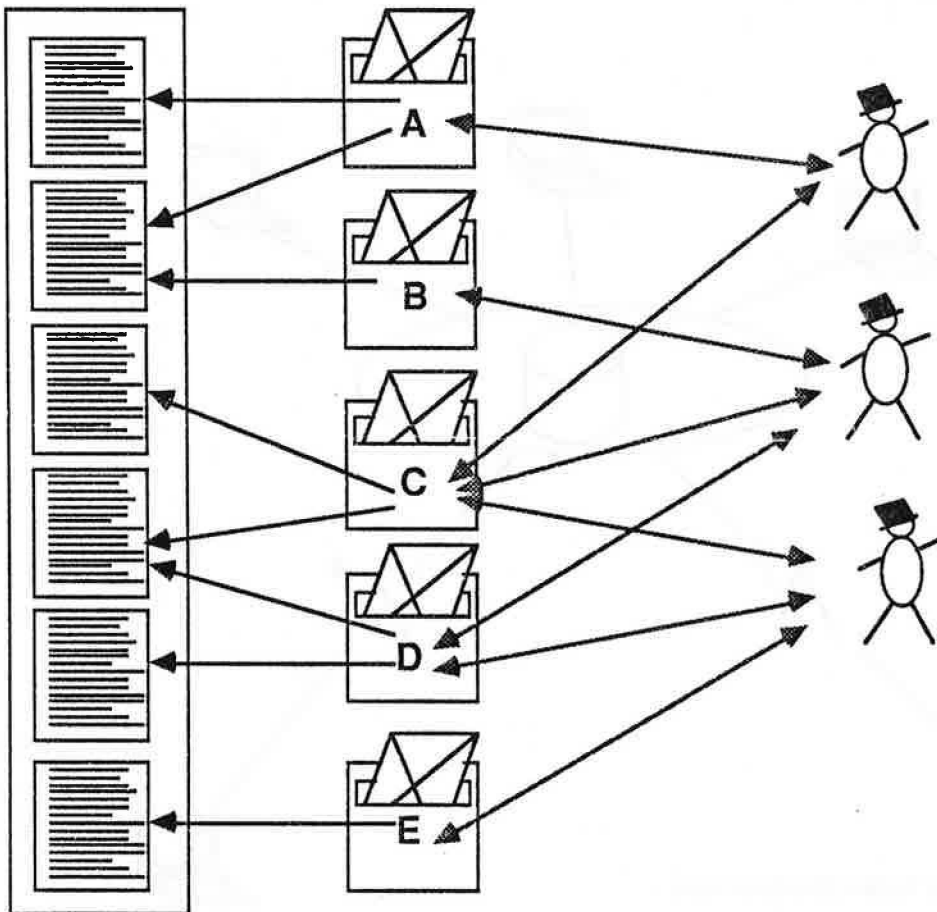
Fig I.1 : Accès au service de conférences



## 2 ETAT DE L'ART

Il existe actuellement bon nombre de systèmes de conférence, parmi eux on peut citer COM à Stockholm qui est détaillé plus loin, EuroKom à Dublin qui sert de moyen de communication pour toute personne impliquée dans un projet Esprit des Communautés Européennes, Forum sur Multics, les *news* d'Unix, et TOOL, un système développé sur des machines IBM. La plupart de ces produits offrent simultanément un service de courrier de personne à personne et un service de conférences.

### 2.1 UN EXEMPLE : COM



Les activités A, B et E servent de boîtes aux lettres. C et D sont des conférences. Un même courrier peut appartenir à plusieurs activités.

Fig. I.2 : COM exemple de système de conférences

Parmi les systèmes de conférences informatiques utilisés aujourd'hui, on remarque COM qui fut développé en Suède en 1977 et qui a été remanié à plusieurs reprises. Il a servi de base à EuroKom. Il gère un ensemble de boîtes aux lettres, une par utilisateur pour le courrier de personne à personne et une par conférence, chacune de ces boîtes représente une activité. Tous les messages du système, qu'ils appartiennent à une conférence ou qu'ils fassent l'objet d'une communication interpersonnelle, font partie d'une ou plusieurs activités. Dans le cas du courrier de personne à personne, les messages contenus dans une boîte ne peuvent être lus que par son unique propriétaire. La boîte d'une conférence est accessible par ses membres pour consulter ou écrire des messages.

Un message entre individus appartient à autant d'activités que de destinataires. De même, un même texte concernant plusieurs conférences appartient à autant d'activités. Dans tous les cas, le texte du message n'est pas dupliqué. L'appartenance à une activité donnée est faite par un lien entre cette activité et le texte du message. Communiquer revient donc à créer un texte, puis à le lier aux activités concernées (Fig. I.2).

## 2.2 CENTRALISATION DU PRODUIT

COM est un outil fortement centralisé. Tous les usagers doivent être connus sur le site de Stockholm et les messages doivent y être stockés. Cela signifie qu'un usager distant doit être connecté au site central pour consulter sa boîte aux lettres. Ceci se révèle très coûteux, surtout lors d'appels à grande distance.

A titre d'extension, COM permet la retransmission de messages vers le site de l'usager distant. Le message est alors envoyé dans la boîte aux lettres de la machine locale de l'usager. Cette opération est moins coûteuse que la consultation distante, mais elle prive l'utilisateur de toutes les facilités de COM, la copie du message envoyée n'appartenant plus à l'environnement de la machine centrale.

## 2.3 GESTION DES UTILISATEURS

COM permet un contrôle des utilisateurs, par exemple tout abonné du système n'a pas implicitement le droit de devenir membre de toutes les conférences. Si n'importe quel utilisateur de COM peut devenir membre d'une conférence dite "ouverte", en revanche, une conférence "fermée" est limitée à certains d'entre eux. De plus, le système permet de différencier les membres d'une conférence donnée. Certains ne peuvent que lire les messages inclus dans une activité, l'ajout de messages leur est interdit.

tion et aussi supprimer toute information litigieuse. De même, tout auteur de messages doit être capable d'ôter ses créations de la conférence. Outre ces considérations juridiques, les systèmes doivent supporter des impératifs commerciaux. Il faut offrir plusieurs types de facturations possibles. Le coût peut être à la charge de l'auteur (petites annonces), du lecteur (journal), de l'organisateur (communication interne à une entreprise). Faire payer la présence de messages dans la conférence implique que les clients aient la possibilité d'ajouter et retirer facilement leurs messages.

#### 4.2.2 Adaptation des conférences

On se rend compte que les conférences sont d'une utilisation différente selon leur champ d'application. Autant dans un milieu de recherche une plus grande liberté d'expression est prévisible, autant dans un environnement commercial un contrôle plus strict est nécessaire. Il faut que le système de conférences soit adaptable à un large éventail d'applications. Les options qu'il doit offrir concernent surtout les droits des abonnés. C'est en jouant sur ces droits que l'organisateur contrôle le flux des informations. Ce qui change également entre différentes utilisations d'un système de conférences, c'est la nature des relations qui existent entre les messages. Pour les petites annonces, des relations comme "réponse à" sont fréquentes. Dans un système décrivant des produits d'une entreprise, des textes "nouvelles versions de" sont utiles. Il faut que selon l'application souhaitée, l'organisateur puisse définir les liens possibles entre messages, les modalités d'attribution des droits des abonnés et le système de facturation du service.

#### 4.2.3 Réduire les coûts

Une plus grande utilisation des outils de communication passe par une réduction des coûts de fonctionnement. Dans un système centralisé comme l'est COM, les consultations sont rapidement trop onéreuses. Les frais seraient moindres si l'architecture était moins centralisée, ainsi, les consultations entraîneraient beaucoup moins de communications à longue distance.

#### 4.2.4 Pour un accès simplifié.

Un impératif à une plus large utilisation des conférences informatiques est d'en faciliter l'usage. Intéressons-nous au cas d'un utilisateur distant qui veut consulter une conférence d'un système donné. Il doit d'abord arriver à se connecter à la machine hôte de la conférence, puis il doit posséder une connaissance minimum du système d'exploitation de cette machine. Lorsqu'enfin il se retrouve dans le système de conférences, il fait face au jeu de commandes qui permettent de se brancher sur une conférence donnée, de lire les textes non lus, de créer des messages ou de

répondre à un message. Une fois qu'il possède une connaissance suffisante des commandes précédentes, l'utilisateur tente des opérations plus complexes telles que créer une conférence, ramener la copie d'un message sur sa machine, rechercher un message d'après des références trouvées dans un texte. Les produits actuels sont trop complexes d'accès pour être destinés à un large public.

De plus, il faut prendre en considération les nouvelles architectures de communication. Les tendances dans le domaine de la bureautique prévoient de fournir à l'utilisateur un accès à différents serveurs, à partir d'un unique poste de travail [ECMA86]. Le service est rendu par une machine à laquelle il accède directement par réseau local ou par un site éloigné atteignable à travers un réseau à longue distance. Dans tous les cas, les opérations de connexion demeurent transparentes à l'utilisateur (Fig. I.3).

Le poste de travail est équipé pour une interface améliorée par rapport à un terminal classique. Il est souvent pourvu d'un écran graphique, d'une souris et d'un système d'exploitation multifenêtre. De plus, il dispose de ressources locales, mémoire vive, mémoire de stockage : disquette, disque dur, ... et imprimante.

## 5 CONCLUSION DE CETTE VUE D'ENSEMBLE

Au fur et à mesure que les moyens de traitement et de communication se développent, les besoins en outils de conférences vont croître. Pour y répondre, il faut spécifier un produit qui gère efficacement les messages, leurs relations ainsi que les membres des conférences. Il faut offrir un accès convivial aux usagers et tenter de réduire les coûts de communication.

The first part of the document is a letter from the Secretary of the State to the Governor, dated January 1, 1900. It contains a report on the work of the State during the year 1899. The letter is signed by the Secretary and is addressed to the Governor.

The second part of the document is a report on the work of the State during the year 1899. It is a detailed account of the various departments and their activities. The report is signed by the Secretary and is addressed to the Governor.

The third part of the document is a report on the work of the State during the year 1899. It is a detailed account of the various departments and their activities. The report is signed by the Secretary and is addressed to the Governor.

The fourth part of the document is a report on the work of the State during the year 1899. It is a detailed account of the various departments and their activities. The report is signed by the Secretary and is addressed to the Governor.

<b>LE SERVICE CRMM</b>	<b>11</b>
<hr/>	
<b>GESTION DE L'INFORMATION</b>	<b>12</b>
Groupes de discussion	12
Relations entre messages	12
Limiter les contraintes	13
Propositions actuelles	14
Gestion des nouveautés	14
Relations entre messages	15
Les travaux en cours	15
<b>TRAVAUX DE GROUPE ASSISTES PAR ORDINATEUR.</b>	<b>16</b>
Influences sur les organisations	16
Concepts de base	17
Le modèle AMIGO	18
Les limites du modèle	19
<b>LES PROPOSITIONS DE CRMM</b>	<b>20</b>
La base des messages	20
Les fiches de CRMM	20
Relations entre fiches	21
Opérations sur les fiches	29
Vision personnalisée de la base	34
Remarques sur l'archivage	34
CRMM exemple de communication de groupe	34
Description de CRMM	35



## Chapitre 2

### LE SERVICE CRMM

Le principal service offert par les systèmes de communication consiste à acheminer et délivrer des messages entre correspondants. Ainsi, le courrier de personne à personne laisse aux soins de l'utilisateur la gestion des messages reçus. Lorsque le nombre des échanges croît, des problèmes d'archivage apparaissent, l'utilisateur ressent alors le besoin d'un filtrage des messages pour déterminer rapidement ceux qui l'intéressent, et d'un stockage efficace pour retrouver des messages précédemment lus.

Les services du courrier de personne à personne ne sont pas suffisants pour prendre en compte toutes les informations qui décrivent les correspondants, notamment leur organisation. Or, les nouveaux supports informatiques qui permettent le travail en groupe, influent énormément sur les organisations qui les utilisent. Les concepteurs doivent être conscients de ces effets, pour les définir et tenter de les maîtriser, ils doivent disposer de modèles qui permettent de décrire, de réaliser et d'exploiter au mieux les nouveaux produits de communication.

Comme le courrier électronique, CRMM met en relation beaucoup d'interlocuteurs, de plus, il considère cet ensemble d'utilisateurs comme un groupe organisé et non pas comme une simple collection d'individus. La bonne collaboration des membres du groupe passe par le contrôle du flux des informations échangées. En outre, le système de conférences permet l'échange de nombreux messages qui doivent être archivés. Ces deux aspects, stockage des messages et gestion du groupe des utilisateurs conditionnent fortement les services offerts par CRMM.



# 1 GESTION DE L'INFORMATION

## 1.1 GROUPES DE DISCUSSION

La communication devient difficile lorsque le nombre d'interlocuteurs et celui des messages augmentent. Or, des systèmes de conférences mettent en relation des nombres importants d'utilisateurs. Pour remédier aux difficultés que cela entraîne, les systèmes doivent faciliter la création de petits groupes de discussion dont les effectifs sont plus réduits. Par exemple, un thème de conférence peut faire naître des conversations entre une partie des membres sur des sujets plus restreints.

Pendant les spécifications de CRMM, deux approches furent envisagées pour favoriser l'éclosion de ces groupes réduits d'interlocuteurs. D'abord, cela peut passer par la création d'une sous-conférence, un membre du groupe crée une conférence interne à celle en cours. Il faut dans ce cas que la création et l'administration soient simples pour les utilisateurs du système. L'autre possibilité consiste à demeurer dans la conférence et à permettre des conversations restreintes en séparant les messages d'une même discussion.

La première solution est séduisante. Toutefois, le contrôle du système devient difficile à assurer. Ceci n'est pas trop grave tant que le système reste centralisé et limité à des milieux spécifiques comme celui de la recherche où la liberté de communiquer est grande. En revanche, la mise en œuvre devient délicate pour des systèmes à large audience pour lesquels des contraintes de facturation et de contrôle de diffusion de l'information sont à prendre en compte. La possible répartition du système ne fait qu'aggraver les difficultés.

## 1.2 RELATIONS ENTRE MESSAGES

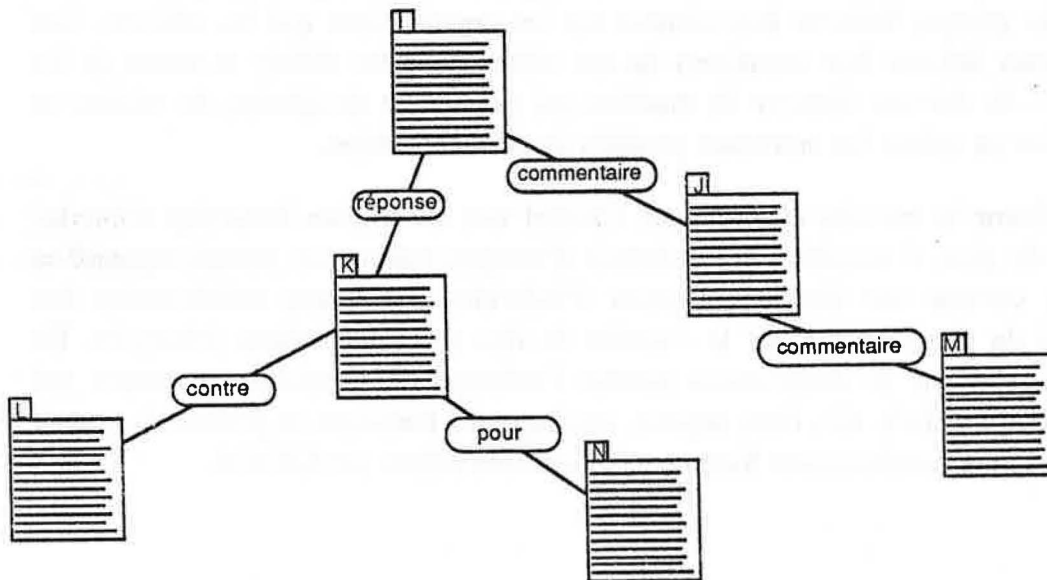


Fig II.1 : Exemple de conversation obtenue dans la base des fiches

Dans tout système de communication, il existe des relations entre les messages échangés (Fig. II.1). En effet, ceux-ci ne sont pas indépendants les uns des autres. Ils se répondent, se commentent, s'annulent ou apportent des précisions. Ces liens entre messages sont riches en information. Par exemple, un message dont le contenu est "oui" n'aura d'intérêt que si la question à laquelle il répond, est connue. Ces informations sur la nature des messages sont très importantes, elles méritent d'être prises en compte par le système. Elles doivent apparaître lors de la lecture, mais peuvent aussi servir pour la sélection d'autres messages.

La nature des relations entre messages varie d'un type de conférence à l'autre. Un système utilisé pour faire connaître les produits d'une entreprise, emploiera par exemple des relations comme "nouvelles versions", "options". Chaque conférence doit disposer de relations propres dont la signification est adaptée au thème qu'elle aborde. Parallèlement, il faut tenir compte du fait qu'une même personne peut être membre de plusieurs conférences. Elle ne doit pas être gênée par des définitions de relations, différentes d'une conférence à l'autre. Pour éviter cela, les relations universelles comme "réponse" sont prédéfinies et ont une signification unique pour toutes les conférences du système. De la même façon, pour une conférence donnée, les relations qui lui sont propres sont acceptées et fixées une fois pour toutes lors de l'initialisation. Ainsi, leur signification acquiert une stabilité nécessaire à une bonne utilisation.

### 1.3 LIMITER LES CONTRAINTES

Ces nouveaux services ne veulent pas engendrer trop de contraintes d'utilisation. En particulier, ils doivent tenir compte des différences entre usagers, que ce soit dans leur expérience du produit, leurs besoins spécifiques ou dans leurs préférences. Ainsi, les critères pour déterminer les messages intéressants ne doivent pas leur être imposés. De plus, même s'ils sont choisis par les usagers, ils sont susceptibles d'évoluer au cours du temps, ils ne doivent donc pas être figés. Pour laisser à chacun le choix de ses critères, les possibilités de filtrage des messages doivent être étendues et les moyens de sélection suffisamment souples et puissants.

Ces systèmes de conférences mettent en relation un grand nombre de personnes. Ceci permet des échanges impromptus qui s'avèrent parfois riches en informations. Ces conversations spontanées naissent souvent lors de flâneries (*browsing*) parmi les conférences. Ce type d'utilisation ne peut guère être amélioré par le système, mais ce dernier doit permettre les consultations au hasard des messages accessibles par le lecteur.

## 1.4 PROPOSITIONS ACTUELLES

Actuellement, les systèmes comme le courrier interpersonnel ou les conférences offrent aux utilisateurs quelques facilités qui répondent aux points évoqués dans les paragraphes précédents. Dans ce qui suit, deux exemples sont donnés. Ils sont généralement disponibles sur les systèmes d'échanges de courrier. Mais ce sont pratiquement les seuls services offerts par la plupart des outils de ce type.

### 1.4.1 Gestion des nouveautés

Il est certain qu'un message récent que l'utilisateur n'a pas encore lu, offre *a priori* plus d'intérêt que les messages plus anciens de la boîte aux lettres. Lorsque l'utilisateur se connecte à la machine hôte, un message lui signale le nombre des nouveaux messages. Au début de la session de travail, l'outil de communication propose en priorité les messages non lus (Fig. II.2), ils sont obtenus par exemple en frappant un simple retour-chariot. Cette facilité n'est pas prise en compte par la normalisation. Il s'agit d'une fonction locale qui est offerte aux utilisateurs par la plupart des produits qui gèrent une boîte aux lettres.

```
Login : brun
passwd :
<-----Bienvenue sur smc----->
mail x400 : 2 nouveaux messages (0..19)

$>_
```

Le nombre de nouveaux messages est signalé ainsi que la numérotation actuelle des messages contenus dans la boîte

Fig II.2 : Exemple de connexion sur une machine hôte d'un système de messagerie

### 1.4.2 Relations entre messages

La messagerie de personne à personne telle qu'elle recommandée par le CCITT, permet à ses utilisateurs de déclarer leurs courriers liés à de précédents messages. Les types de relations disponibles sont :

- en réponse à,
- référence,
- annule.

Ce sont les seuls types de relations entre messages qui sont actuellement recommandés par le CCITT. Ils vont sûrement évoluer suite aux travaux décrits ci-après.

### 1.5 LES TRAVAUX EN COURS

Des études sont menées actuellement sur ces problèmes d'archivage de messages par les différentes institutions normalisatrices (CCITT, ISO, AMIGO, ECMA). Leur finalité principale est d'étendre les services du courrier de personne à personne et de spécifier des serveurs d'archivage normalisés qui offriront leurs fonctionnalités à toutes sortes d'applications. Ils seront utiles par exemple, pour gérer les boîtes aux lettres du courrier de personne à personne, ou pour archiver les messages des conférences. Outre les applications de communication, ils serviront au stockage de documents purement locaux.

A propos de documents, il faut citer les travaux qui portent sur cette notion. Lorsque la nature des documents aura été spécifiée et que des outils permettront de les manipuler en respectant une norme bien établie, les services de courrier ou de conférence pourront en tirer profit pour améliorer la qualité des messages échangés.

Ces thèmes sont repris dans le chapitre sur la messagerie. Compte-tenu des délais de réalisation de ces produits, CRMM fonde son étude sur les spécifications et outils disponibles. Il définit donc un service d'archivage qui lui est propre.

## 2 TRAVAUX DE GROUPE ASSISTES PAR ORDINATEUR.

Ce titre de paragraphe provient de l'expression CSCW pour *Computer Supported Co-operative Work* qui a reçu ses lettres de noblesse lors de la conférence d'Austin de Décembre 1986.

### 2.1 INFLUENCES SUR LES ORGANISATIONS

Toute activité qui met en scène plusieurs entités (personnes, processus automatisés,...) nécessite des communications entre ses intervenants. Ce qui est décrit ici ne s'intéresse pas aux tâches réalisées dans le cadre de cette activité, mais aux échanges d'informations qu'elle réclame. L'ensemble des membres du groupe est vu comme un **groupe de communication**, on parle alors d'une **organisation**. On peut donner comme exemple les entreprises, les services à l'intérieur d'une entreprise ou une équipe formée à l'occasion d'un projet.

Le travail à l'intérieur d'une organisation est étroitement lié au flux des informations [FELDMAN86]. Par exemple, celui-ci influe énormément sur la coopération interne à un groupe de travail. Le débit et la structure de la communication modifient la connaissance de l'organisation, les contacts entre ses membres, la collecte de l'information et donc les décisions.

Les nouveaux moyens de communication offerts par l'informatique bouleversent profondément les flux d'informations. Traditionnellement, les communications internes à une organisation ont une tendance naturelle à se restreindre à des interlocuteurs rapprochés au sens spatial (même bureau) ou structurel (même équipe). Cette limite provient de l'ignorance réciproque de membres éloignés qui peuvent pourtant avoir des intérêts communs. Elle est aujourd'hui dépassée grâce aux nouveaux outils comme le courrier électronique. Bien sûr, celui-ci permet la communication entre usagers qui se connaissent préalablement, mais en plus, il facilite les rencontres entre nouveaux interlocuteurs. Cette capacité provient des caractéristiques suivantes :

- Le mode de communication asynchrone demande moins de coordination de la part des interlocuteurs (ils n'ont pas à être disponibles au même instant).
- Les listes de diffusion permettent d'atteindre un ensemble d'individus sans trop d'efforts supplémentaires.
- Les messages sont stockés, ils sont accessibles tant qu'ils ne sont pas détruits. Contrairement à un outil comme le téléphone, le courrier électronique conserve une trace des messages.

Maintenant, il faut tirer parti de ces nouveaux outils de communication. Ils permettent déjà des services classiques comme le courrier et ceux-ci sont perfectibles. Mais, il serait bien qu'ils servent également pour des activités reposant actuellement sur des supports non informatiques plus coûteux, moins rapides, moins faciles d'emploi et moins efficaces. Ces progrès passent par une étude préalable de l'activité bénéficiaire, en particulier des schémas de communication qu'elle implique.

## 2.2 CONCEPTS DE BASE

Quel que soit le médium employé, tout **processus de communication** met en jeu un ensemble d'interactions entre individus dans le but d'échanger de l'information. Tout d'abord, la compréhension de cette information nécessite un langage commun aux correspondants.

La connaissance de ce langage n'est pas suffisante à une bonne communication. La réception de messages, le mot est pris dans son sens le plus général, déclenche de nouveaux échanges. Cette production de messages suit des **règles** spécifiques au processus de communication concerné. Celles-ci dépendent fortement des interlocuteurs. Un même message ne provoque pas toujours des réactions identiques, l'effet dépend des individus impliqués dans la communication. Ces différences de réactions distinguent des classes parmi les intervenants du processus de communication. Ces classes sont dues aux **rôles** différents joués par les intervenants. De plus, l'effet d'un message est conditionné par l'historique du processus, selon les messages précédents, la réception d'un message entraîne des actions variées. Tout processus de communication repose donc sur :

- des échanges d'information,
- des règles,
- des rôles pour les intervenants.

La bonne connaissance de ces paramètres par les intervenants est nécessaire pour une bonne évolution du processus. Soit ils sont définis explicitement comme c'est le cas dans les jeux de cartes, ils peuvent alors être enseignés aux nouveaux correspondants. Soit, et c'est le plus fréquent dans la vie courante, ces paramètres sont acquis par observations et essais. Ces paramètres proviennent de lois dont le respect est indispensable au fonctionnement du processus (imposées par la technologie ou la juridiction), mais ils prennent également en compte des règles d'usage qui constituent un véritable **savoir-vivre** ou une **étiquette** propre au processus. Par exemple, la présentation des textes (citations, paragraphes, justification, titres) peut faire l'objet de telles recommandations.

De plus, ces paramètres ne sont pas figés, ils évoluent au cours du déroulement du processus. Leur évolution fait partie de la définition du processus.

Les processus de communication ne sont pas parfaitement autonomes. Par exemple, avant leur mise en place, des échanges d'informations sont nécessaires, ne serait-ce que pour fixer les rôles et les règles. Il convient donc de préciser les éventuelles relations avec d'autres processus.

Les contraintes amenées par les règles ne sont pas de la même importance pour tous les processus. Ceux qui supportent un jeu de cartes sont fortement régulés. Les échanges d'informations au cours d'une partie de bridge sont réglementés. En revanche, le simple courrier de personne à personne ne suit que peu de lois. Or, les futurs produits de communication (vote, prise de décision, gestion de rendez-vous..) actuellement à l'étude, tendent à être de plus en plus exigeants. Les activités distribuées qu'ils vont supporter vont imposer le respect de règles donc de protocoles contraignants.

Pour espérer définir des outils de ce type, il faut disposer de modèles qui permettent de les décrire, de les simuler et de les réaliser. Leur intérêt dépasse le stade de la conception et du développement, ils sont utiles en phase d'exploitation pour que chaque usager comprenne le fonctionnement de l'activité et le sens des rôles qu'il joue.

### 2.3 LE MODELE AMIGO

Le groupe de travail AMIGO est en train d'élaborer un modèle de description de groupe. Les éléments de cette proposition sont donnés ci-dessous. Cette étude n'est pas encore définitive, mais ce qui suit reflète bien les éléments principaux pour modéliser un outil de communication de groupe.

**Activité :** Tout groupe se forme autour d'un projet commun à ses membres. L'activité du groupe tend vers ce but. L'édition d'un document est un exemple d'activité partagée par un ensemble d'individus.

**Rôles :** Un rôle correspond à un ensemble de tâches qu'un usager doit effectuer. Chaque intervenant est amené à jouer un ou plusieurs rôles, réciproquement, un rôle peut demander la coopération de plusieurs individus.

**Fonctions :** Elles décrivent les actions qui génèrent ou traitent les messages. Par exemple, la création, la destruction ou la réponse sont des fonctions du courrier de personne à personne.

**Descripteur de messages :** Toutes les informations qui doivent figurer dans les

messages sont données dans le descripteur. Il donne le type du message, sa structure, le codage utilisé, l'historique du message, en fait tous les traitements subis et les éventuelles relations avec d'autres messages du processus.

**Règles :** Les règles précisent les fonctions licites sur les messages compte tenu du type des messages et des rôles des correspondants.

**Attribution des rôles :** Ceci décrit comment les rôles sont assignés aux usagers. En particulier, si un rôle est obligatoire ou optionnel, si le nombre des usagers qui peuvent le remplir est limité ou si un rôle est implicitement jouable par tous les intervenants.

## 2.4 LES LIMITES DU MODELE

La communication humaine est complexe et diverse (culture, profession, éducation). Il paraît difficile de vouloir la modéliser dans son ensemble. Aussi, les ambitions des travaux tels que ceux menés par le groupe AMIGO, se limitent à des communications structurées et organisées avant même d'envisager un support informatique. De plus, il considère seulement les communications qui se rapportent à une activité donnée : production d'un document, gestion de rendez-vous, vote...

Ce modèle doit permettre une description standardisée de toute communication de groupe qui s'inscrit dans les limites précédentes. Il a été bâti à partir de l'expérience d'outils existants ou au moins envisageables actuellement. Il ne faut pas qu'il impose des limites à des projets futurs qui s'en accommoderaient mal. De plus, un outil doit rester flexible et adaptable aux conditions d'utilisation, même si sa description respecte le modèle.

En outre, ce modèle ne doit pas amener une bureaucratisation excessive pour les concepteurs et pour les futurs utilisateurs des produits qu'il permet de décrire. Ainsi, l'usager doit garder une importante marge de manœuvre. Lorsqu'il communique, toutes les réactions et productions de messages qui s'en suivent, n'ont pas à être automatisées.

Enfin, les outils de communication ont des effets politiques et sociaux. Ils doivent respecter certaines règles sur la vie privée et sur la liberté individuelle des usagers. Il faut donc respecter les lois en vigueur, mais compte tenu du retard du législateur face au progrès technique, les concepteurs du modèle et des applications qui le respectent, doivent avoir en tête ces soucis éthiques.



## 3 LES PROPOSITIONS DE CRMM

### 3.1 LA BASE DES MESSAGES

Le besoin de gérer les messages conduit à construire une base structurée de messages. Elle prend en compte des attributs sur les messages et elle offre des opérateurs de sélection. Elle permet de définir les relations entre messages.

Désormais, l'entité composée par un message et les attributs qui le caractérisent est appelée **fiche**, le message en est le **corps**. Dans ce paragraphe, nous décrivons cette entité et les opérations qu'elle supporte.

#### 3.1.1 Les fiches de CRMM

Outre le corps, une fiche contient dans son **en-tête** les informations sur le message (Fig. II.3). La liste qui suit n'est pas close, d'autres attributs seront définis dans la suite de la thèse.

- Un identifiant. Il est unique et généré par le système.
- Les attributs sur la fiche :
  - Le nom de l'auteur.
  - La date de prise en compte du message par la conférence.
  - La durée de validité de la fiche. Elle est donnée par l'auteur ou fixée par défaut par le système. Lorsque la durée est fournie par l'auteur, elle ne peut en aucun cas être supérieure à une valeur connue du système.
  - L'objet du message. Il est donné par l'auteur de la fiche. Généralement, il donne un court résumé ou un titre au contenu du message.
  - Les mots clés, eux aussi sont fournis par l'auteur.
- Les liens sur d'autres fiches. Cette notion est approfondie dans le paragraphe suivant.

Actuellement, le contenu ou corps du message est purement textuel. Des versions futures de CRMM permettront d'échanger des informations de nature variée telle que les images, les graphiques, la voix... De même, le contenu est simple, il n'est pas structuré. Par exemple, il ne contient pas d'autres fiches. Cette amélioration passe par l'inclusion du corps d'un message du courrier de personne à personne pour bénéficier de sa structure et de son codage.

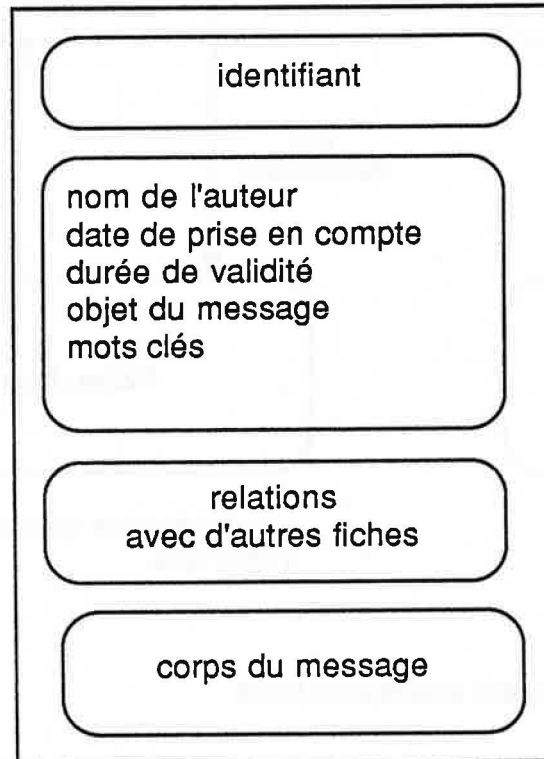


Fig II.3 : Structure d'une fiche CRMM

### 3.1.2 Relations entre fiches

#### 3.1.2.1 Liens entre fiches

Lorsqu'une fiche est créée, il est possible de définir des relations vers des fiches déjà présentes dans la conférence. Il est également possible de relier des fiches existantes de la base. Ces relations sont de différentes natures et impliquent des comportements particuliers. Elles sont décrites par des liens entre fiches.

Ces liens sont orientés. Leur sens respecte l'ordre d'apparition des fiches dans la base lorsqu'ils sont définis à la création. Sinon, il est fixé par celui qui les ajoute entre des fiches déjà présentes. Créer une relation génère en fait une paire de liens. Un est dit **lien ascendant** entre une fiche et sa précédente. L'autre dit **lien descendant** est de sens opposé (Fig. II.4).

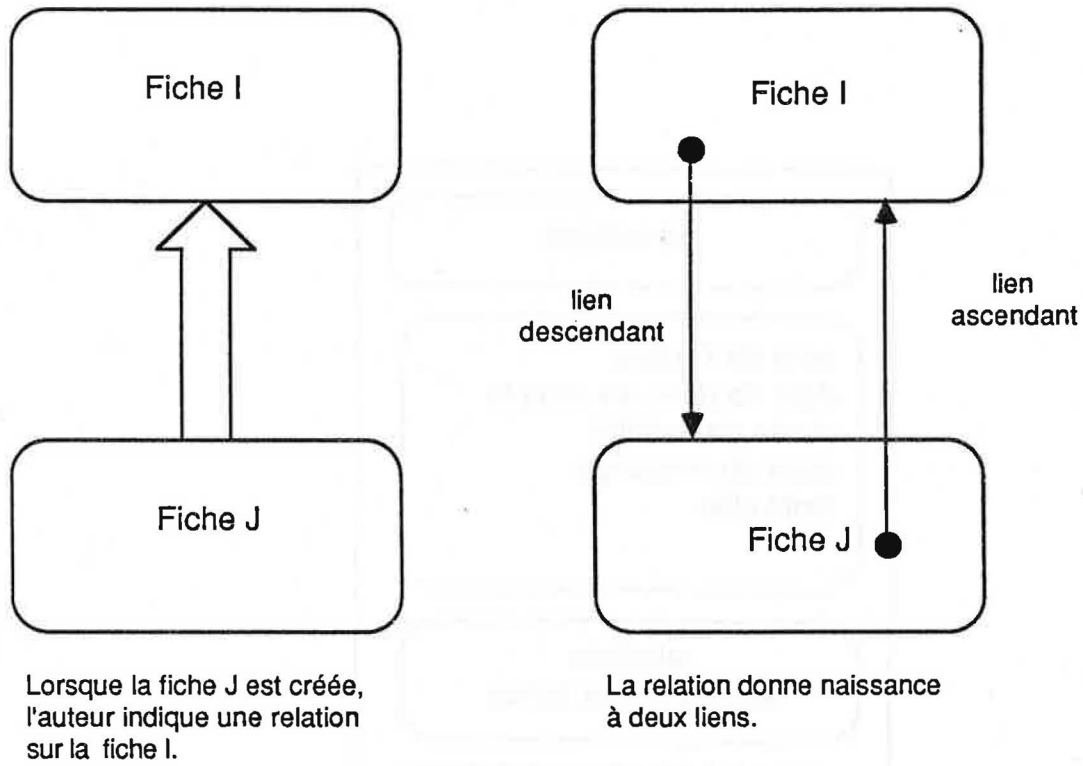


Fig. II.4 : Relation et liens entre fiches

Ces liens structurent la base des fiches. Le modèle de base obtenu diffère selon le nombre de liens ascendants autorisés. Ainsi, si une fiche est liée à une fiche précédente au maximum, on obtient un ensemble d'arbres non connectés. Dans CRMM, une même fiche peut être liée à plusieurs fiches précédentes. La structure obtenue est alors un ensemble de graphes. De plus, il ne peut exister entre deux fiches qu'une relation au plus (Fig. II.5).

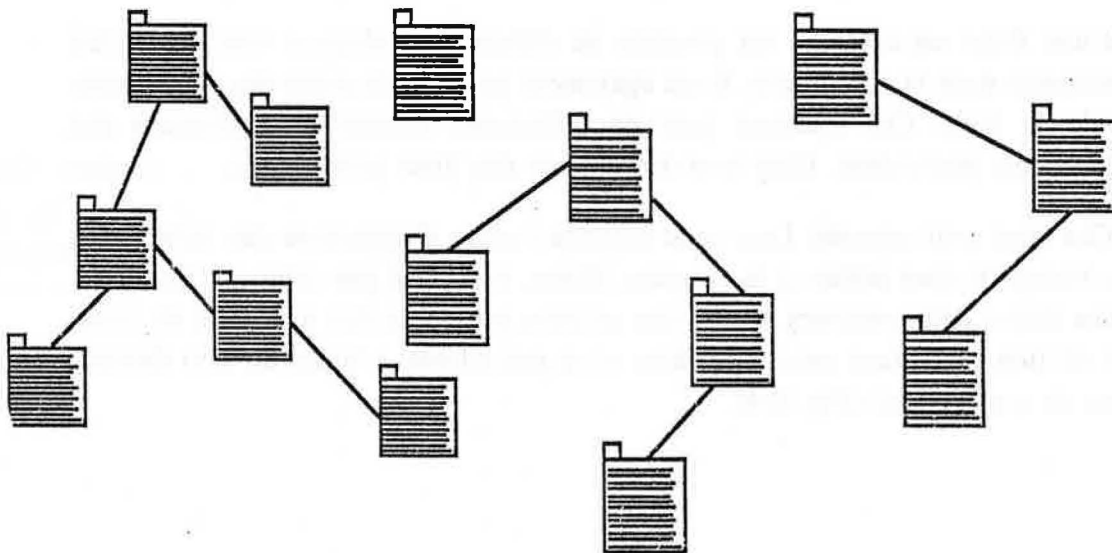


Fig II.5 : Structure de la base des fiches

Chacun des graphes non connectés apparaît comme une conversation dont l'effectif est plus réduit et le sujet est plus restreint (Fig. II.6). Les liens favorisent la création de petits groupes de discussion plus adaptés à une communication efficace. De plus, cela permet de relier les fiches d'un même sujet, ce qui offre un moyen de filtrer les messages. En ne lisant que les messages appartenant à un graphe donné, l'usager limite sa lecture à des fiches supposées l'intéresser.

Les relations entre fiches sont de nature différente, aussi, quatre classes de liens ont été définies. Chacune d'elles inclut différents types. Certains d'usage général sont prédéfinis, d'autres peuvent être propres à une conférence donnée. Les opérations sur les fiches induisent des effets dépendant des relations qu'elles ont avec d'autres.

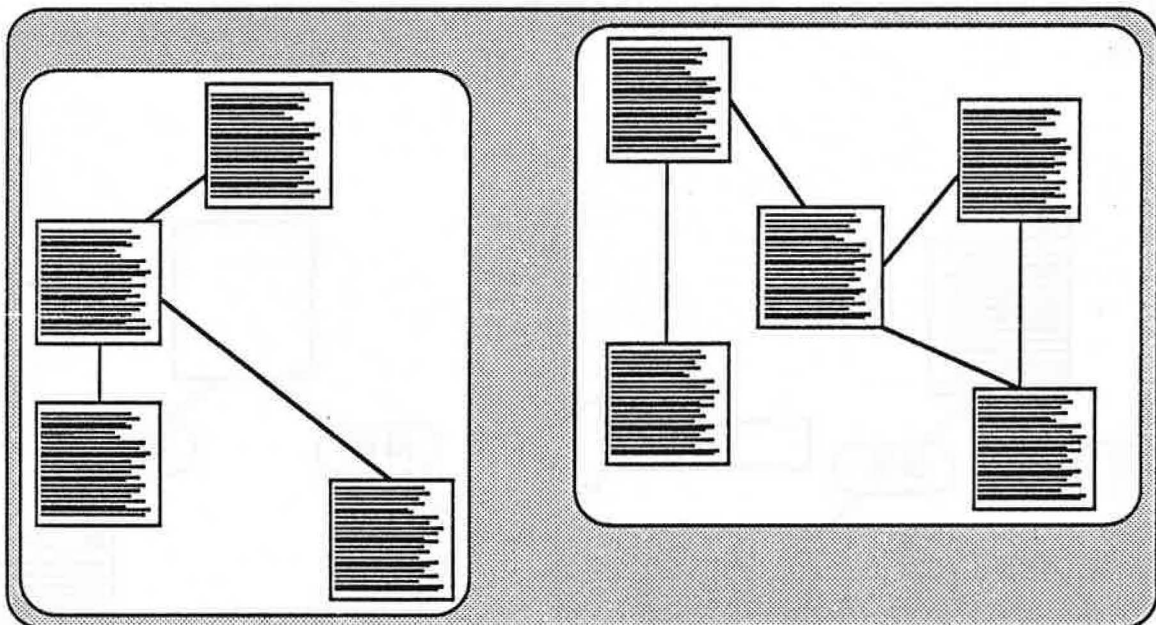
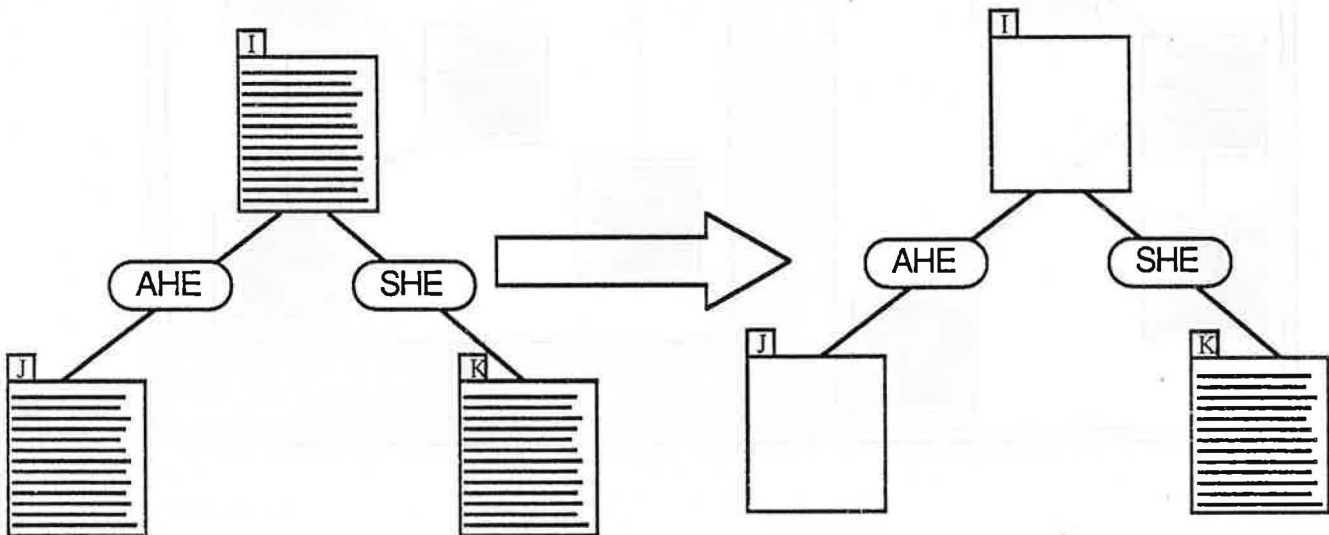


Fig II.6 : Mise en évidence des conversations dans la base des fiches

### 3.1.2.2 Liens séquentiels

Les liens séquentiels sont donnés par les auteurs de fiches. Ils expriment une relation avec des fiches déjà présentes dans la base. L'ordre de lecture des fiches qu'ils lient respecte la chronologie d'ajout dans la base. Le type de lien *réponse à* en est un exemple. Ils permettent au lecteur de reconstituer la chronologie des messages et de suivre les conversations qu'ils ont créées.

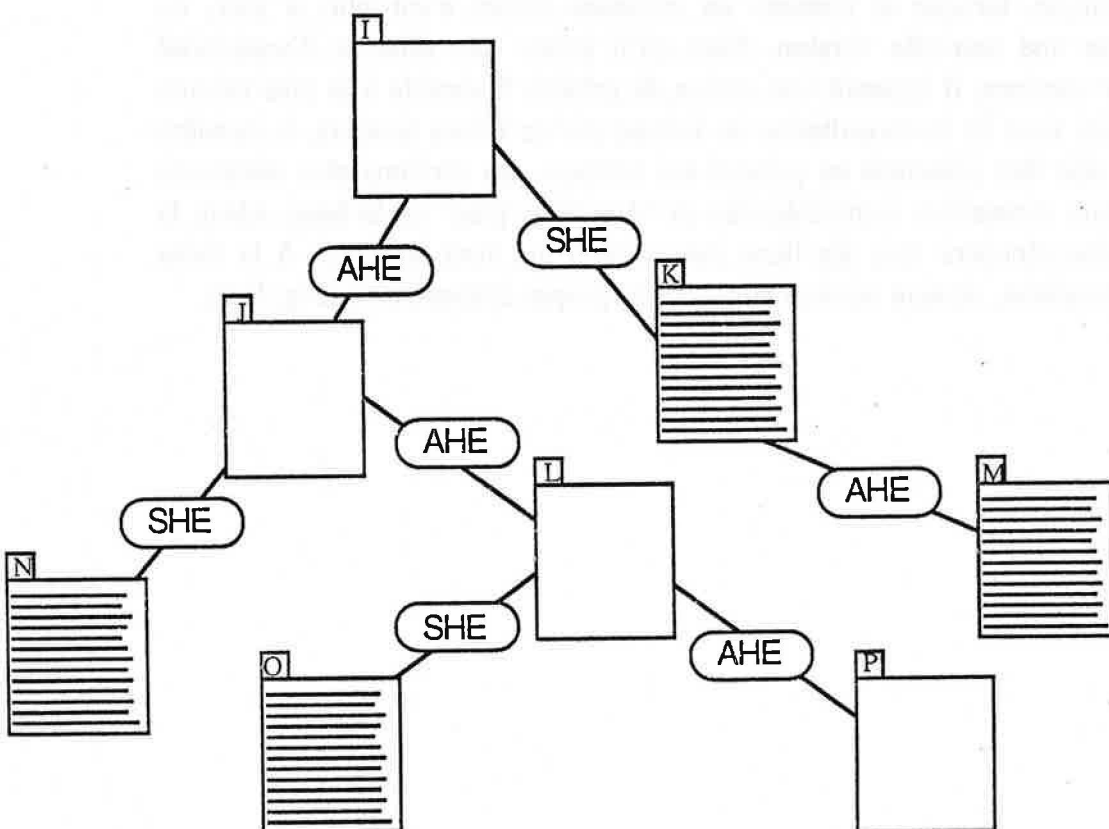
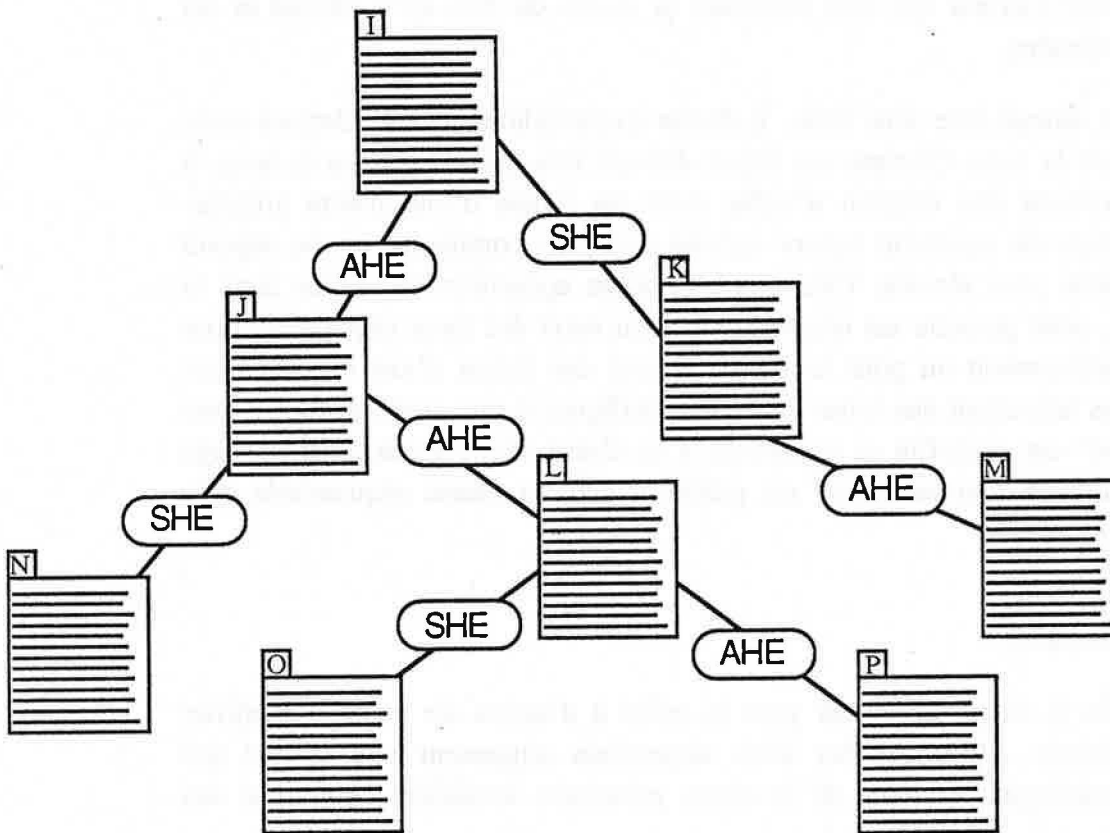
Parmi ceux-ci, CRMM distingue deux classes selon les effets qu'ils provoquent sur les fiches. Si la seconde fiche perd tout sens lorsque la fiche antécédente est effacée ou détruite, le lien doit prendre en compte cette forte dépendance. Comme cela, la disparition de la première fiche efface également la seconde devenue sans intérêt. En revanche, si la seconde fiche conserve sa valeur malgré l'effacement de la première, elle ne doit pas disparaître. Dans le premier cas, le lien choisi est de la classe "séquentiel avec héritage de l'effacement : AHE", dans le second cas, il s'agit d'un lien "séquentiel sans héritage d'effacement : SHE" (Fig. II.7).



AHE : lien de classe Avec Héritage de l'Effacement  
SHE : lien de classe Sans Héritage d'Effacement

Fig II.7 : Héritage de l'effacement limité au lien de classe AHE

Lorsqu'une fiche est effacée par héritage, sa descendance subit elle-aussi ce traitement tant que le type des liens l'autorise (Fig. II.8).



II : Le service CRMM

Fig II.8 : Transmission de l'effacement à toute une descendance

Cette notion d'héritage est intéressante pour le lecteur, car elle permet un nettoyage automatique de la base en ôtant des fiches devenues sans intérêt, mais elle est contraignante pour l'auteur qui doit connaître la classe du lien qu'il choisit et les effets qu'elle engendre.

Lorsqu'un auteur crée une fiche, il donne éventuellement des relations avec d'autres fiches de la base. Comme ces fiches doivent être présentes dans la base, il existe nécessairement une relation d'ordre entre les fiches d'une même arborescence. Le principe de causalité assure qu'une fiche ne commente ou ne répond jamais à une fiche plus récente. Ceci empêche toute apparition de cycles dans la base des fiches, cette garantie est nécessaire lors du suivi des liens séquentiels pour l'héritage de l'effacement ou pour la simple lecture des fiches d'une descendance. Ces deux classes admettent des types de liens spécifiques à une conférence. Le type de lien "réponse" est prédéfini et appartient à la classe séquentielle avec héritage d'effacement, le type "commentaire" est prédéfini dans la classe séquentielle sans héritage d'effacement.

### 3.1.2.3 Lien prioritaire

A la création de la fiche, un auteur peut la relier à d'autres de façon à modifier l'ordre des lectures. Alors que les liens séquentiels supposent une lecture qui respecte la chronologie, les liens de la classe prioritaire modifient la priorité des consultations.

Par exemple, lorsque le contenu de certaines fiches n'est plus à jour, un usager propose une nouvelle version. Bien qu'il existe une relation d'antériorité entre les deux versions, il apparaît une notion de priorité favorable à la plus récente des deux fiches. Lors de la consultation de la base par de futurs lecteurs, la dernière fiche en date doit être présentée en priorité aux lecteurs, les versions plus anciennes quoique toujours accessibles sont reléguées au "deuxième plan" de la base. Ainsi, la nouvelle version récupère tous les liens descendants qui font référence à la fiche périmée. En revanche, chaque version conserve sa propre descendance (Fig. II.9).

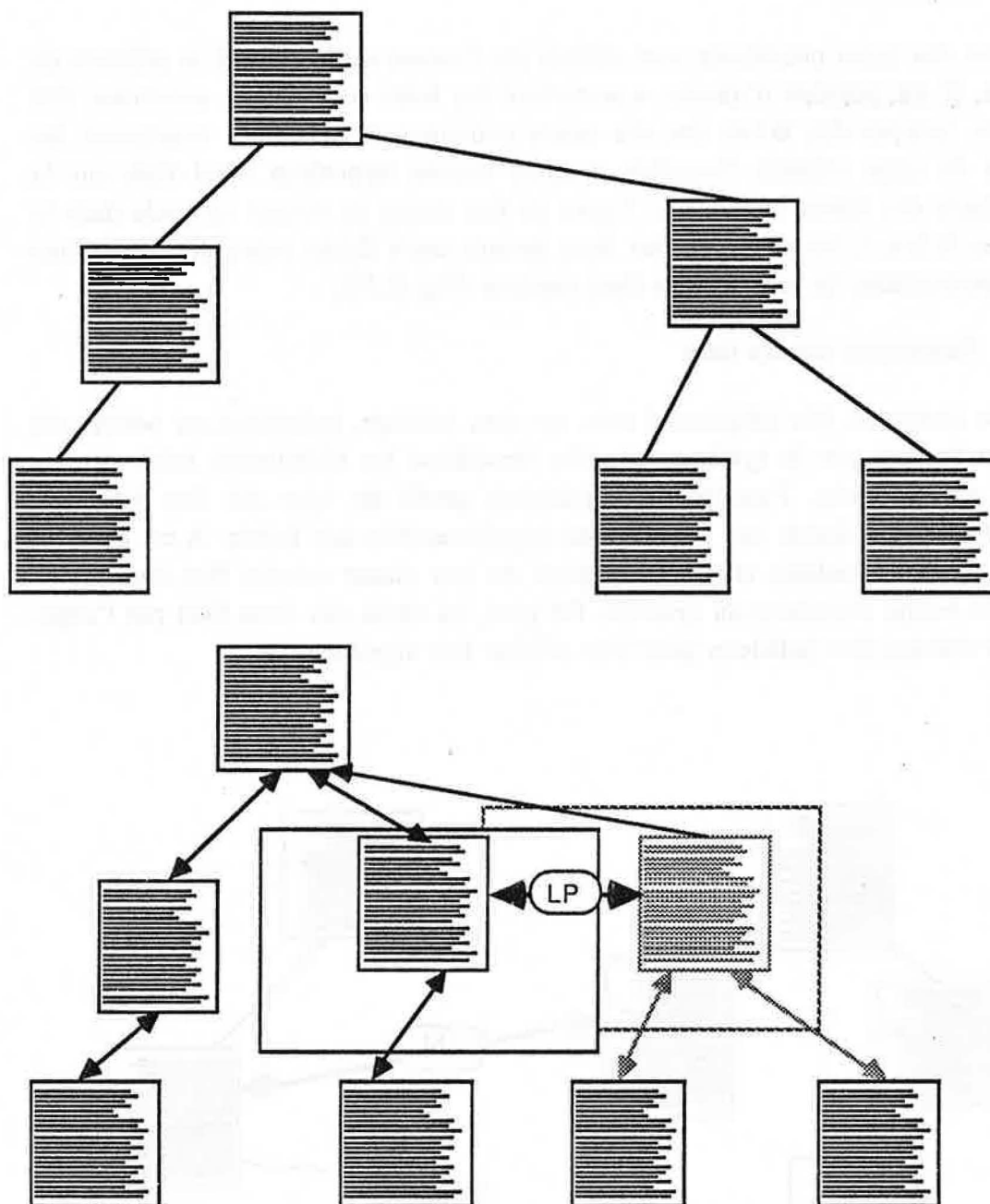


Fig II.9 : Exemple de lien prioritaire entre deux fiches



### 3.1.2.4 Liens neutres

Les liens des types précédents sont définis par l'auteur de la fiche à la création de la fiche. Il est possible d'ajouter *a posteriori* des liens entre fiches existantes. Par exemple, lorsque des fiches ont des sujets communs de tels liens avertissent les lecteurs de cette affinité. Toutefois, comme aucune hypothèse n'est faite sur la chronologie des fiches impliquées, l'ajout de lien risque de fermer un cycle dans la base des fiches. C'est pourquoi, les liens ajoutés entre fiches existantes sont d'une classe particulière, ils sont appelés liens neutres (Fig. II.10).

### 3.1.2.5 Remarques sur ces liens

Seule la classe du lien (séquentiel avec ou sans héritage, prioritaire ou neutre) est prise en compte par le système, car elle caractérise les traitements subis par les fiches. En revanche, l'utilisateur tire également profit du type du lien (réponse, commentaire) qui donne une signification supplémentaire aux fiches. A cet effet, la liste de liens disponibles et une description de leur classe doivent être disponibles pour une bonne utilisation du système. De plus, les choix des liens faits par l'organisateur doivent être judicieux pour bien refléter leur signification.

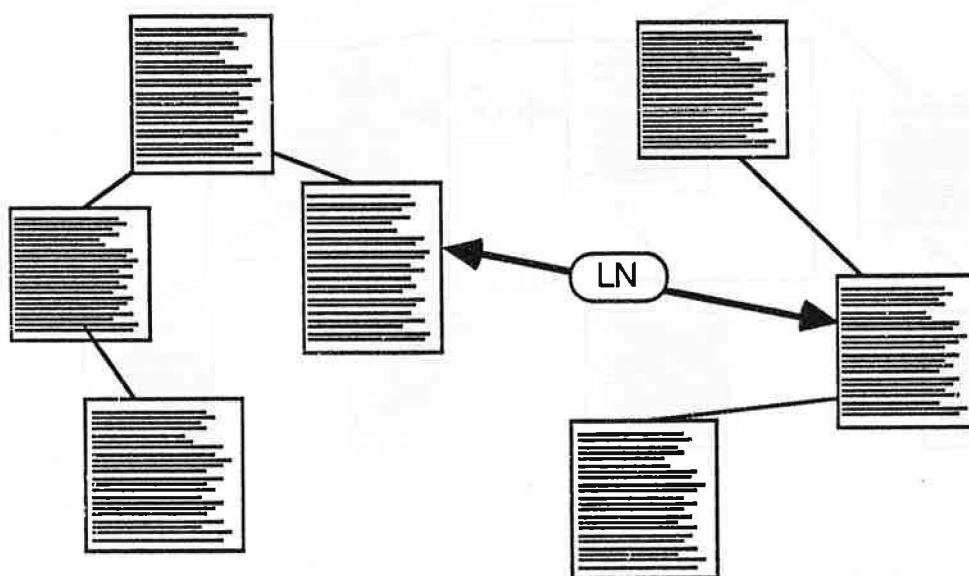


Fig II.10 : Exemple de lien neutre

### 3.1.3 Opérations sur les fiches

#### 3.1.3.1 Création

Lors de la création d'une fiche, l'auteur doit préciser l'objet, les mots clés et les éventuels liens sur d'autres fiches de la base. Il peut proposer une durée de validité.

#### 3.1.3.2 Ajout

Lors de l'ajout d'une fiche, les relations qu'elle définit sont répercutées sur les fiches concernées et un identifiant est généré, la fiche est datée et la durée de validité fixée.

#### 3.1.3.3 Sélection

L'ensemble des fiches est organisé en une base de données. Ceci autorise des opérations de sélection sur cette base. Les traitements comme la lecture, l'effacement, la destruction de fiches, l'ajout ou la suppression de liens peuvent porter soit sur une fiche désignée par son identifiant, soit sur un ensemble défini par des critères de sélection. Par exemple, un administrateur local peut demander l'effacement de toutes les fiches antérieures à une date donnée.

Dans un souci de généralisation du traitement des requêtes, la désignation par l'identifiant est vue par le système comme une sélection dégénérée à un seul critère.

#### Les attributs

Les sélections portent sur les attributs de l'en-tête définis précédemment :

- l'identifiant,
- le nom de l'auteur,
- la date d'introduction,
- la date de péremption,
- l'objet,
- les mots-clés,
- les relations avec d'autres fiches.

Outre les attributs contenus dans la fiche, la sélection peut tenir compte du fait que la fiche a été ou non, lue par le demandeur.

### Les opérateurs

Selon les critères, différents opérateurs sont possibles. Ainsi, les attributs ordonnés comme les dates ou les numéros locaux acceptent les opérateurs de comparaison.

Les critères forment des **chaînes de sélection** plus ou moins complexes. Pour cela, ils se combinent au moyen d'opérateurs booléens :

- ET,
- OU,
- NON.

Ceci donne comme exemples de chaînes de sélection possibles :

Chercher les fiches dont l'auteur est "brun" et qui sont dans la base depuis le 20 avril 1987.

*auteur = "brun" et date de prise en compte < "20-04-87"*

Chercher les fiches portant sur APL et qui répondent à la fiche 235.

*mot clé = "APL" et lien ascendant = "réponse:235"*

L'intérêt de la sélection est de pouvoir manipuler des ensembles de fiches. Ceci est agréable lorsque la conférence contient énormément de messages.

#### 3.1.3.4 Lecture

La lecture de toutes les fiches de la conférence est autorisée à tous les membres de la conférence. La désignation des fiches à lire se fait par sélection, en donnant un identifiant ou implicitement en demandant une fiche non lue.

### 3.1.3.5 Effacement

L'effacement d'une fiche n'affecte que son corps. Elle demeure présente mais elle est vide.

La transmission de l'effacement est récursive. Une fiche qui reçoit l'effacement de son ascendance la transmet à sa propre descendance toujours en respectant la classe des liens. Le fait que les graphes n'aient pas de cycles assure que la propagation de l'effacement ne risque pas de boucler indéfiniment. Garder une fiche vide est utile car la fiche privée de corps conserve quand même des informations comme les relations. Les supprimer ferait perdre contact entre des fiches d'un même graphe (Fig. II.11). Une fiche qui possède plusieurs ascendants directs est effacée si tous ses parents sont effacés ou détruits, et si tous les liens supportent l'héritage de l'effacement.

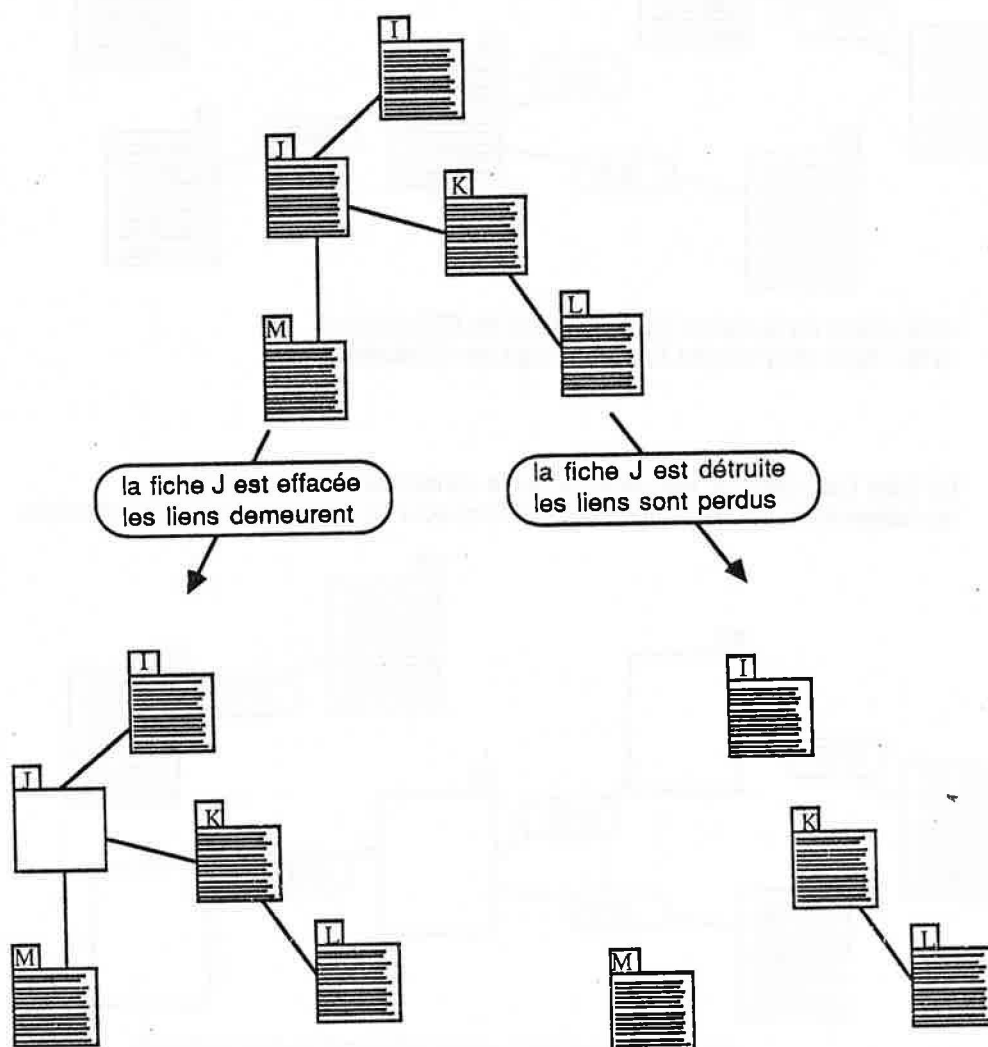
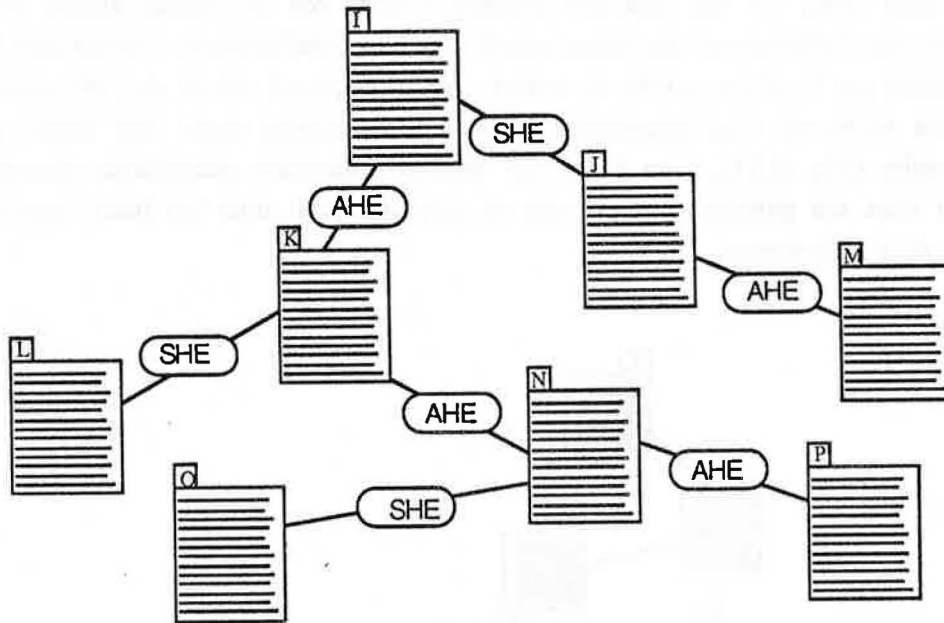


Fig II.11 : Fiche vide et fiche détruite

### 3.1.3.6 Destruction

La destruction supprime entièrement la fiche, corps et en-tête. Elle a la même influence sur la descendance de la fiche concernée que l'effacement (Fig. II.12). Elle rompt les relations dans lesquelles la fiche est impliquée. Cet effet est répercuté sur les fiches directement liées.



AHE : liens de la classe Avec Héritage de l'Effacement  
SHE : liens de la classe Sans Héritage de l'Effacement

La fiche I est détruite, les relations qui la concernent, disparaissent, les fiches de la descendance dont les liens sont de la classe AHE sont effacées,

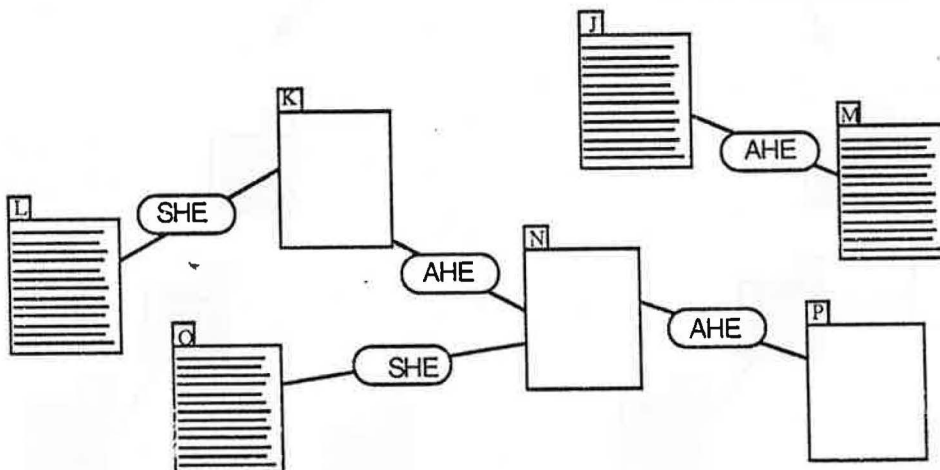


Fig II.12 : Effet de la destruction sur la descendance

### 3.1.3.7 Ajout de liens

Il est possible d'ajouter des liens neutres entre des fiches déjà existantes. Ils ne transmettent pas l'effacement. Ainsi, ils ne risquent pas de fausser la structure de la base. Cette opération modifie les deux fiches concernées. Aucune autre relation directe ne doit les lier.

### 3.1.3.8 Suppression de lien

La suppression des liens ne pose pas ce genre de problèmes. Il est possible de supprimer n'importe quel lien (Fig. II.13).

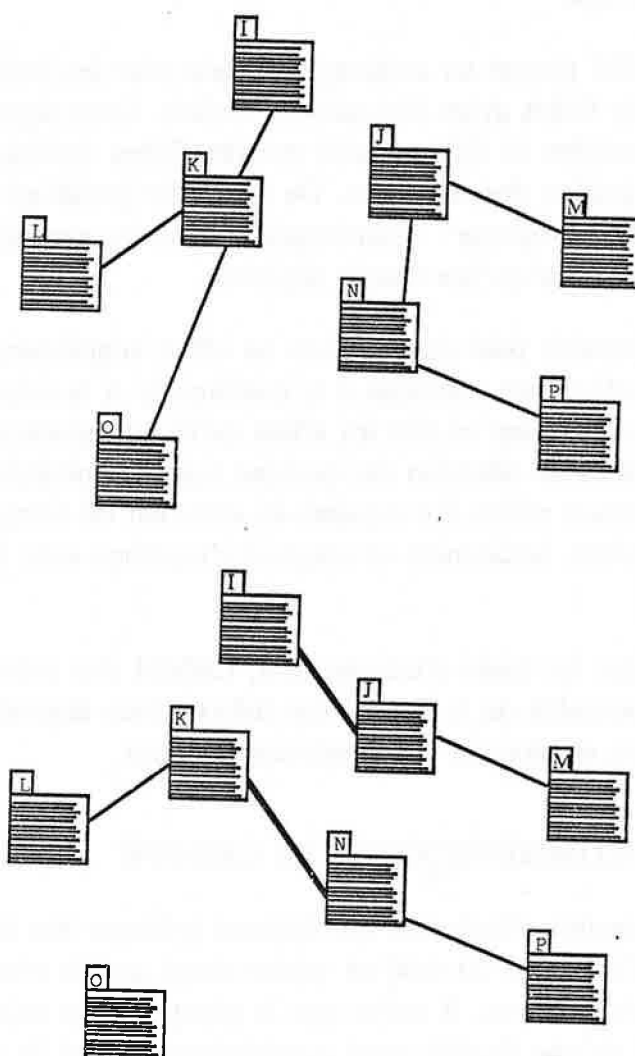


Fig II.13 : Ajout et suppression de liens

### 3.1.4 Vision personnalisée de la base

Pour chaque utilisateur, le système gère une vision de la base qui lui est propre. Ainsi, le système CRMM tient le compte des fiches qu'il n'a pas encore lues. Lorsque l'utilisateur se présente, le système peut lui en indiquer le nombre et lui permettre d'y accéder rapidement et simplement.

Lorsqu'un usager repère une fiche intéressante, il peut la marquer pour la retrouver aisément par la suite. La valeur de la marque est choisie par l'utilisateur et lui est personnelle. Cela lui permet de nommer les fiches de façon plus mnémotechnique que la désignation par identifiant.

### 3.1.5 Remarques sur l'archivage

La base de messages de CRMM permet un archivage efficace pour les lecteurs, ils peuvent filtrer ou retrouver des fiches ayant des attributs choisis. Cette organisation permet de manipuler des ensembles de fiches plutôt que des fiches isolées, ce qui est agréable pour assurer la gestion des messages. De plus, elle prend en compte des relations comme "prioritaire", "réponse", "commentaire" entre les messages pour constituer une structure plus élaborée qu'une simple séquence.

Toutefois, cette base demande pour être efficace un effort supplémentaire de la part des auteurs qui ont à relier leurs messages à la conférence. A la création, ils doivent choisir le ou les liens, en ayant en tête les effets qu'ils produisent sur leur message. En outre, les opérations de sélection sur quelque base de données que ce soit, ne sont pas simples, la nature même des requêtes de sélection est complexe. Il faut veiller à faciliter ces requêtes, notamment en soignant l'interface avec l'utilisateur.

Enfin, comme pour toutes les bases d'informations, CRMM doit prévoir une fonction d'administrateur responsable de la qualité des informations disponibles. Il peut agir en effaçant des fiches, en ajoutant ou supprimant des liens.

## 3.2 CRMM EXEMPLE DE COMMUNICATION DE GROUPE

Un groupe CRMM est un ensemble d'individus qui désirent échanger des informations sur un sujet commun. Ce groupe est créé en même temps que la conférence par l'abonnement des premiers membres. Il arrive que le groupe existe auparavant et que la conférence ne soit qu'une facilité pour communiquer. C'est le cas des groupes de travail qui se servent des conférences comme moyen de communication entre les réunions. Malgré l'arrivée ou le départ des membres, le groupe continue d'exister. D'ailleurs, pour la plupart des conférences d'une certaine taille, tous les membres ne se connaissent pas nécessairement.

Le système CRMM distingue différents usagers. Pour permettre un contrôle des informations ajoutées à la conférence, seuls certains d'entre eux ont le droit d'écrire des messages directement dans la base. Ce sont les **écrivains**. Ceux qui ne disposent pas de cette facilité sont appelés **auteur**, ils doivent soumettre leur création à un **éditeur** qui accorde ou non l'autorisation d'ajout. Cette requête est connue sous le nom de **demande d'imprimatur**. Le type de la conférence fait varier la proportion d'usagers autorisés à écrire directement. Ainsi, une conférence dans un milieu scientifique le permettra à tous ses membres, en revanche, l'organisateur d'une conférence utilisée comme moyen de publication plus officiel, exigera des auteurs une soumission préalable à un comité de lecture. Cette possibilité est nécessaire pour les publications où la responsabilité de l'organisateur est engagée. Il faut lui permettre un contrôle sur les écrits de la conférence.

Dans le groupe des usagers, il faut prévoir un **administrateur** pour abonner les membres, leur allouer ou non des droits. A cette gestion des usagers, s'ajoute la gestion des messages de la base. Dans le chapitre précédent, des opérations d'entretien de la base ont été présentées. Par exemple, il faut supprimer les messages périmés ou relier des fiches. Cette gestion de la conférence est elle-aussi à la charge de l'administrateur.

### 3.3 DESCRIPTION DE CRMM

CRMM a été choisi comme un exemple dans la définition du modèle AMIGO, la description est donnée dans [DANIELSEN86]. Toutefois, il faut remarquer les problèmes de redondance qui ne manquent pas d'apparaître. Il est en effet difficile de définir des rôles sans en même temps donner les règles qui leur sont attachées. Les règles caractérisent également les fonctions disponibles dans le système.

La description qui suit ne respecte pas une syntaxe particulière. Elle présente les éléments qui participent à la définition de CRMM. Le système est vu comme un ensemble de rôles qui dialoguent avec un rôle particulier : le service de conférences (Fig. II.14). Chaque rôle est détaillé en donnant les messages émis et reçus, les règles suivies, les fonctions traitées. Enfin, les descripteurs de chaque type de message sont définis. Ils en donnent les constituants. Implicitement, chaque message contient l'identification de l'émetteur et du destinataire.



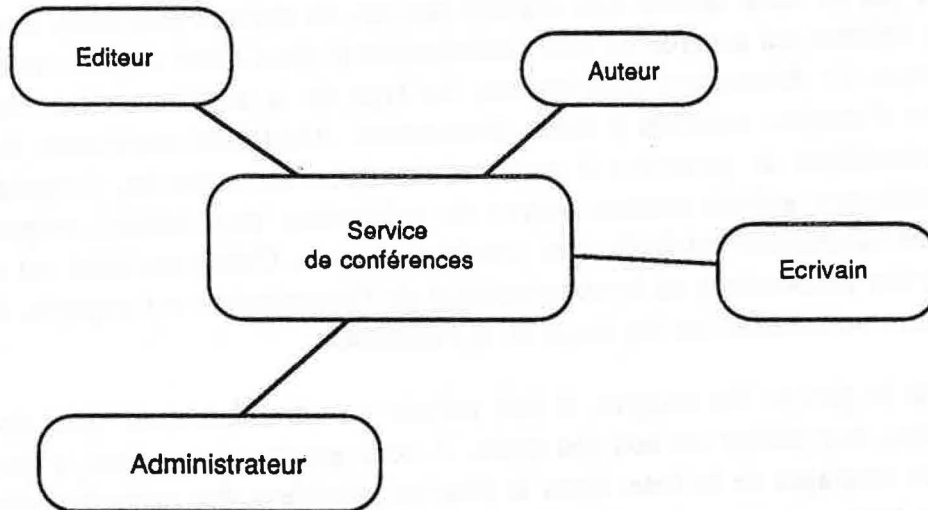


Fig II.14 : Rôles de CRMM

## Les rôles

### lecteur

messages reçus :

- fiches pour consultation

messages émis :

- demande de lecture

fonctions :

- marquage

règles :

- la lecture est autorisée pour toutes les fiches

instanciation :

- tout abonné est lecteur

**auteur**

messages reçus :

- notification de réponse d'imprimatur
- réponse d'opération (effacement)

messages émis :

- demande d'imprimatur
- demande d'effacement

fonctions :

- création de fiche

règles

- l'auteur peut choisir l'éditeur auquel il soumet la fiche
- l'auteur peut demander l'effacement de ses fiches
- l'auteur peut proposer une date de péremption pour sa fiche

instanciation :

- tout usager non écrivain est auteur

**écrivain**

messages émis :

- fiche pour ajout
- demande d'effacement de fiches

messages reçus :

- réponse à ajout
- réponse à effacement

fonctions :

- création de fiche

règles

- l'écrivain peut demander l'effacement de ses fiches

**éditeur**

messages émis :

- réponse d'imprimatur

messages reçus :

- notification de demande d'imprimatur

## fonctions

- répondre à demande d'imprimatur

## instanciation :

- un éditeur est écrivain

**service de conférence**

## messages reçus :

- demande d'opération (destruction, effacement de fiches, ajout, suppression de liens)
- demande d'imprimatur
- réponse d'imprimatur
- abonnement

## messages émis :

- compte-rendu
- notification

## fonctions :

- insertion
- sélection
- effacement
- destruction
- ajout de lien
- suppression de lien

## règles :

- demande d'opération →
  1. si droits conformes alors opération
  2. envoi(réponse d'opération, demandeur)
- demande d'imprimatur →
  1. si pas d'éditeur alors éditeur ← éditeur par défaut
  2. envoi(notification de demande d'imprimatur, éditeur)
- réponse d'imprimatur →
  1. si accord alors insertion de la fiche
  2. envoi(notification de réponse d'imprimatur, auteur)
- abonnement →
  1. modification du répertoire des usagers (abonnement, désabonnement, changement de droits)
  2. envoi(réponse d'opération, administrateur)

**administrateur**

messages reçus :

- réponse d'opération

messages émis :

- demande d'opération
- abonnement

instanciation :

- un seul utilisateur joue ce rôle

**les descripteurs de messages**

**fiches pour consultations :** fiche complète

**fiche pour ajout :** sans identifiant, sans les dates de prise en compte et de péremption

**demande d'imprimatur :** fiche pour ajout et un possible nom d'éditeur

**réponse d'imprimatur :**

- réponse
- commentaire
- fiche soumise

**demande de lecture :**

**demande d'effacement :**

**demande de destruction :**

- identifiant de fiche concernée.

**demande d'ajout de lien :**

**demande de destruction de lien:**

- identifiant des deux fiches concernées

**réponse à opération :**

- nature de l'opération
- compte-rendu

**abonnement :**

- nature de l'opération (nouvel abonnement, changement de droits, désabonnement)
- usager concerné
- nouveaux droits (sauf pour désabonner)

## REPARTITION DE LA CONFERENCE

41

---

<b>EXEMPLES DE CONFERENCES DISTRIBUEES</b>	<b>42</b>
Expérience Cosac	42
TOOL	43
<b>LA SOLUTION CRMM</b>	<b>45</b>
Modèle avant Répartition	45
Liste de diffusion	46
Répartition du répertoire	46
Répartition du rôle d'administrateur	46
Duplication de la base des messages	47
Synchronisation des messages	48
Désignation	51
Désignation des fiches	51
La désignation des usagers.	52
Format des fiches	53
Réception de messages erronés	53
Les messages entre sites	54
Diffusion d'opérations sur la base	54
Message de services	54
Imprimatur	55
Messages d'erreur	55
<b>INTEGRATION DANS LA MESSAGERIE X400</b>	<b>56</b>
La messagerie	56
Fonctions d'un système de messagerie	58
La messagerie X400	59
Description générale	59
Représentation du modèle MHS	63
Syntaxe X409	64
La messagerie COSAC	65
CRMM et la messagerie	66
Conférences et courrier interpersonnel	67
Description des entités	69
Description en couche	70
Désignation	70
Le protocole	72
Primitives de BBGA offertes aux BBUA	75
Primitives de BBUA offertes à l'utilisateur	76
Quelques scénarii	77
Consultation de la base	77
Ajout d'une fiche par un écrivain	77
Ajout d'une fiche par un usager non écrivain	79
Effacement d'une fiche	81



## Chapitre 3

### REPARTITION DE LA CONFERENCE

Le coût des communications est un des principaux obstacles à la banalisation des nouveaux outils qui permettent des échanges d'informations. Ceci est d'autant plus sensible lorsqu'il s'agit de communications internationales. Dans le cas du système COM, l'utilisateur distant paie le temps de connexion à une machine située à Stockholm ou alors le transfert de chacun des messages pour les lire sur sa machine. Pour une plus grande utilisation d'un système de conférence, il convient de limiter ces frais de fonctionnement. Une possibilité consiste à dupliquer sur plusieurs sites la base des messages et à répartir l'ensemble des utilisateurs sur ces sites, afin que la plupart d'entre eux soient locaux à un site. Comme ceci, les consultations seront généralement locales. En revanche, toute modification sur un site est diffusée à l'ensemble des autres sites. Le calcul suivant justifie ce choix.

Pour une conférence de deux cents membres, le nombre de lectures de chaque message est estimé à trois cents et le nombre d'utilisateurs distants dépasse les deux tiers du nombre total. Donc, plus de deux cents échanges à longue distance sont prévisibles pour chaque message [PAYS87].

Lorsque la base des messages est dupliquée sur une dizaine de sites, le taux d'utilisateurs distants tombe alors à dix pour cent. Chaque message créé sur un site demande neuf messages pour être diffusé puis une trentaine de transferts sont nécessaires pour les usagers qui ne sont locaux à aucun des systèmes coopérants.

Globalement, le nombre de communications à longue distance est divisé par cinq. Cela justifie l'idée d'une copie intégrale de la base des messages sur un ensemble de sites coopérants. Outre les gains financiers, cette approche répond à un autre problème. En effet, elle facilite l'accès à la conférence pour les usagers qui ont alors à faire avec un système local mieux connu et n'ont plus à mettre en œuvre



des réseaux à longue distance, avec leurs techniques d'adressage et de passerelles.

## 1 EXEMPLES DE CONFERENCES DISTRIBUEES

### 1.1 EXPERIENCE COSAC

Cette expérience [HUITEMA85] avait pour but d'interconnecter les conférences locales existantes dans diverses institutions scientifiques, telles que l'ENSMSE, l'INRIA ou le CNET. Les machines de ces organisations, des SM90 sous UNIX et des Multics, ont leur propre outil de conférences locales (*news* et *forum*), un logiciel supplémentaire se charge simplement de diffuser les créations locales à l'ensemble des sites. Chacun d'eux possède la liste de diffusion de ces sites et dispose d'une boîte aux lettres spéciale où le logiciel de communication trouve les messages contenant les fiches créées à l'extérieur. Celles-ci sont alors introduites dans la conférence locale selon la procédure habituelle.

Quoique simple cet outil a rendu d'importants services. Pour preuve, il était utilisé par une quarantaine d'institutions. Il permettait à un usager d'atteindre une large audience, chaque message est en moyenne lu par quatre à cinq cents personnes. De plus, tout ceci était possible grâce à un outil de faible coût et local, donc d'emploi plus simple et mieux connu.

La large utilisation de cet outil a mis en évidence les difficultés liées à cette structure de communication. Le manque de protocole pour gérer la diffusion cause tant de troubles que l'administration du système est passée d'une équipe du CNET au centre informatique de ce même CNET, pour aboutir finalement à une société privée.

Parmi les problèmes rencontrés, la gestion de la liste de diffusion montre bien le manque de moyens pour une bonne coopération des sites. Dès qu'un des sites change (ajout ou retrait), cela nécessite le report manuel de la modification sur la liste de chacun des autres sites. De même, la diffusion des messages est tellement discrète que les usagers l'oublent, leurs textes font référence à des personnes, événements, ou objets locaux à leur site, ce qui rend leurs messages incompréhensibles pour les utilisateurs des autres sites.

En outre, le contrôle de l'information varie suivant les organisations. L'absence de régulation et de contrôle par le système, autorise tout abonné à diffuser l'information vers l'extérieur. De plus, il n'existe aucun moyen de contrôler *a posteriori* l'information, par exemple, ôter de l'ensemble des conférences interconnectées, les textes litigieux. D'ailleurs, ces carences dans le contrôle des informations échangées ont amené un important site à se retirer du système.

## 1.2 TOOL

**TOOL** est un système de conférences issu des laboratoires IBM. Son développement a été signalé par quelques articles récents, notamment [CHESS87]. Initialement, il s'agissait d'un simple serveur de fichiers qui permettait l'accès partagé à des outils logiciels. Il fut amélioré pour supporter une discussion à grande échelle entre les développeurs de l'IBM-PC. Il est aujourd'hui largement utilisé (3000 auteurs, 20000 lecteurs) et couvre d'autres sujets de discussion.

Le système de conférence est un réseau de nœuds interconnectés (Fig. III.1). Les messages sont groupés en conférences. Chaque nœud peut demander à avoir une copie active de l'une d'elles. C'est-à-dire que les modifications qu'elle subit sur un autre nœud lui sont communiquées. Lorsqu'un nœud reçoit une demande de message, il répond directement s'il possède l'information, sinon il fait appel à un nœud détenteur.

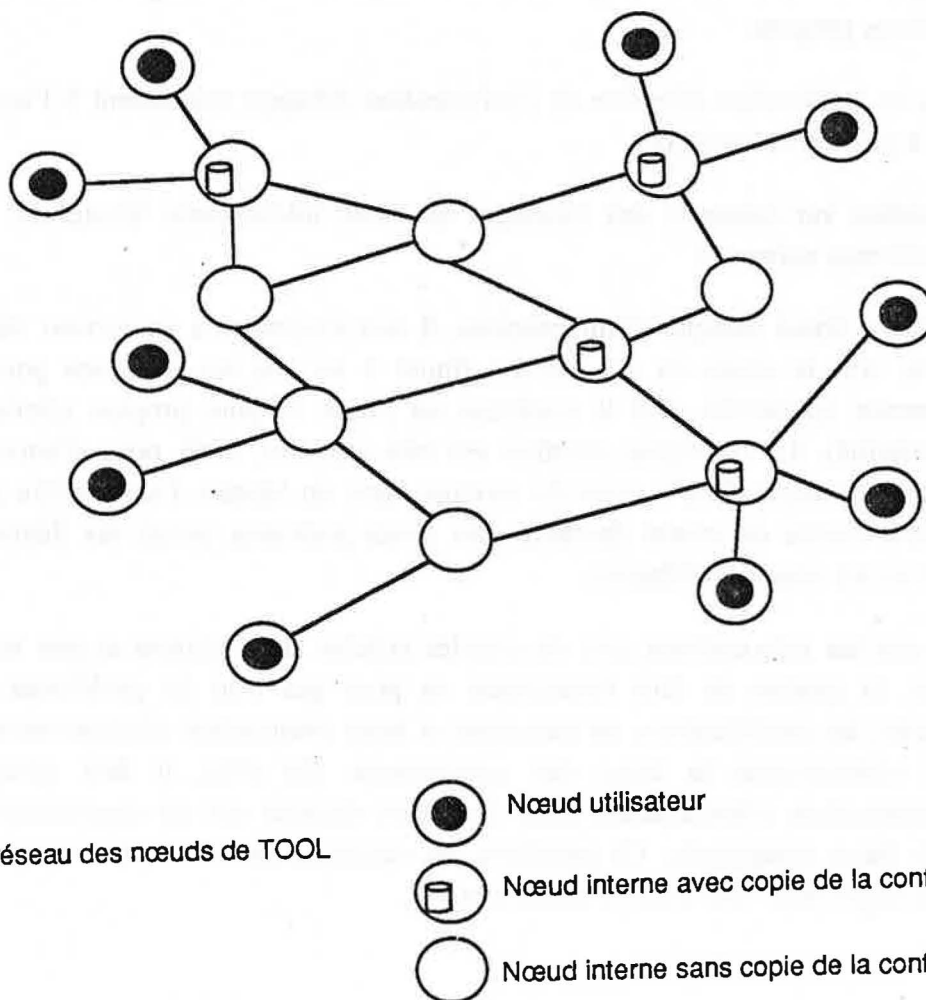


Fig III.1 : Le réseau des nœuds de TOOL

Ce système est séduisant pour plusieurs raisons.

- Il autorise les copies d'informations qui réduisent les coûts de transmission.
- Il permet de choisir les thèmes intéressants, donc de limiter l'encombrement et les messages de mises à jour. Tous les nœuds ne possèdent pas toutes les conférences en copie.
- Il propose une architecture client-serveur. Un même nœud interne au réseau peut être serveur pour les informations qu'il détient et client pour les messages qui lui manquent.
- L'architecture se prête bien à un accès à partir d'un poste de travail autonome pour les usagers du système. Ceci permet d'offrir une interface agréable, performante et personnalisée.
- De plus, ces usagers finals qui accèdent aux nœuds de la périphérie sont toujours clients mais leur nœud peut stocker des messages dans leurs ressources propres.
- Enfin, la localisation effective de l'information échappe totalement à l'utilisateur qui n'a pas à s'en occuper.

Cette duplication sur demande des messages est donc intéressante. Toutefois, elle pose les problèmes suivants :

- Lorsqu'un nœud manque d'informations, il doit s'adresser à un serveur mieux pourvu, soit le choix du serveur est limité à un site de référence pour la conférence concernée, soit il s'adresse au nœud le plus propice (distance, disponibilité). La deuxième solution est très attirante, mais pose d'énormes problèmes comparables à ceux du routage dans un réseau. La première plus simple à mettre en œuvre demande des choix judicieux parmi les thèmes à copier et les sites de référence.
- Tant que les informations sont de simples articles sans relation et non modifiables, la gestion de leur duplication ne pose pas trop de problèmes. En revanche, les modifications de messages et leurs éventuelles relations compliquent sérieusement la tâche des concepteurs. En effet, il faut gérer la synchronisation d'événements issus des lieux distants qui ne communiquent pas de façon instantanée. Ce problème est classique en informatique répartie, il sera repris plus loin dans le document.

## 2 LA SOLUTION CRMM

La solution retenue dans CRMM est proche de celles présentées précédemment. Quoique les études aient été menées indépendamment, CRMM et TOOL préconisent tous les deux une coopération entre différents sites. Chacun de ces derniers dispose de l'intégralité de la base. Ainsi, les utilisateurs travaillent avec un système local. Les modifications sur un site sont diffusées à l'ensemble des autres sites.

Cependant, CRMM se distingue par un contrôle accru du flux des informations et par une base de messages mieux structurée par les relations entre fiches. Nous verrons comment ces deux aspects, subissent les effets de la répartition. Un autre atout de CRMM est de permettre une collaboration entre systèmes hétérogènes en s'appuyant sur les travaux de normalisation. Cette caractéristique concerne autant les machines et les systèmes d'exploitation des sites coopérants que les réalisations de CRMM. Pour peu que les implantations respectent le protocole et offrent les services décrits par le modèle, elles sont capables de collaborer, quels que soient le système d'exploitation, la machine ou l'outil utilisé pour gérer la base des messages.

### 2.1 MODELE AVANT REPARTITION

Les membres de la conférence ne jouent pas tous les mêmes rôles. Ils disposent de droits différents. Si tous peuvent lire, l'insertion de messages est réservée à certains d'entre eux. Le contrôle des droits s'appuie sur un répertoire où chaque usager est inscrit. Cette liste décrit les droits de chacun, à savoir s'il est écrivain, éditeur ou administrateur.

Avant sa répartition, le système comprend la base des fiches et le répertoire des usagers (Fig. III.2).

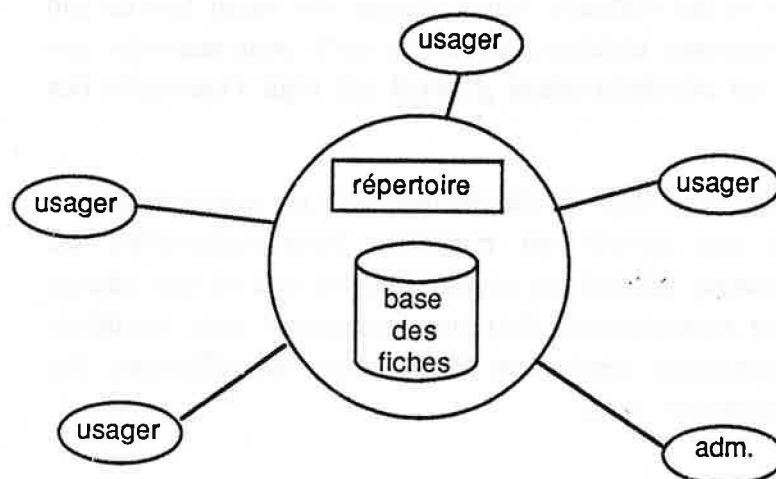


Fig III.2 : Modèle avant répartition

## 2.2 LISTE DE DIFFUSION

Pour communiquer les modifications intervenues localement, les sites doivent connaître les sites coopérants. Ces informations sont comprises dans une liste de diffusion qui donne les désignations des sites et les droits qu'ils peuvent accorder. A chaque modification, intégration, suppression d'un site, modification des droits accordables ou initialisation, cette liste est communiquée à l'ensemble des sites.

## 2.3 REPARTITION DU REPERTOIRE

Comme l'ensemble des membres d'une conférence est réparti sur un ensemble de sites, le répertoire est lui aussi distribué. Ainsi, chacun d'eux dispose d'une sous-liste où sont répertoriés les membres locaux. Le contrôle des droits se fait seulement là où est connu l'utilisateur concerné. Lors de l'insertion d'une fiche par exemple, le droit du demandeur est vérifié localement. Une fois acceptée, la fiche est diffusée sans plus de contrôle des droits de l'écrivain. Lorsqu'il reçoit un message, chaque site vérifie que le site expéditeur appartient bien à sa liste de distribution, mais lui fait confiance pour ce qui est des droits de l'utilisateur. Ceci est compréhensible en contexte distribué, sinon il faut que chaque site ait connaissance de tous les répertoires locaux.

## 2.4 REPARTITION DU ROLE D'ADMINISTRATEUR

L'abonnement des usagers est sous la responsabilité d'un administrateur. Il attribue les droits aux membres de la conférence. Comme chaque site dispose de son propre répertoire, un **administrateur local** doit en être responsable.

Au niveau global, il existe un contrôle des sites. En effet, tous n'ont pas le droit d'accueillir des écrivains ou des éditeurs. Ainsi, chaque site reçoit lors de son intégration ou en cas de changement ultérieur, les droits qu'il peut accorder aux utilisateurs locaux. Il subsiste un **administrateur général** qui régit l'ensemble des sites.

L'administrateur général gère la liste de distribution, elle est communiquée à l'ensemble des sites chaque fois qu'elle est modifiée. Pour authentifier ses messages, le site de l'administrateur général est connu. Il arrive que ce site change en cours d'exploitation. Chaque administrateur doit alors enregistrer cette modification. Le site où officie l'administrateur général est nommé : **site de référence**. Sur celui-ci, il joue le rôle d'administrateur local.

L'administrateur général fournit également le nom d'un éditeur qui est utilisé par défaut lorsque les auteurs non écrivains n'en indiquent pas. Chaque administrateur local se contente de ce nom ou en propose un pour les demandes issues de son site. Il peut choisir un éditeur local pour limiter les coûts de transmission.

## 2.5 DUPLICATION DE LA BASE DES MESSAGES

Pour que la consultation des messages reste d'un faible coût, la base des fiches est dupliquée sur différents sites. Toutes les modifications locales à un site doivent être diffusées à l'ensemble des autres. Par exemple sur le site A (Fig. III.3), un usager ajoute une fiche. Après vérification des droits de l'utilisateur, un message comportant la nouvelle fiche est émis vers l'ensemble des sites coopérants.

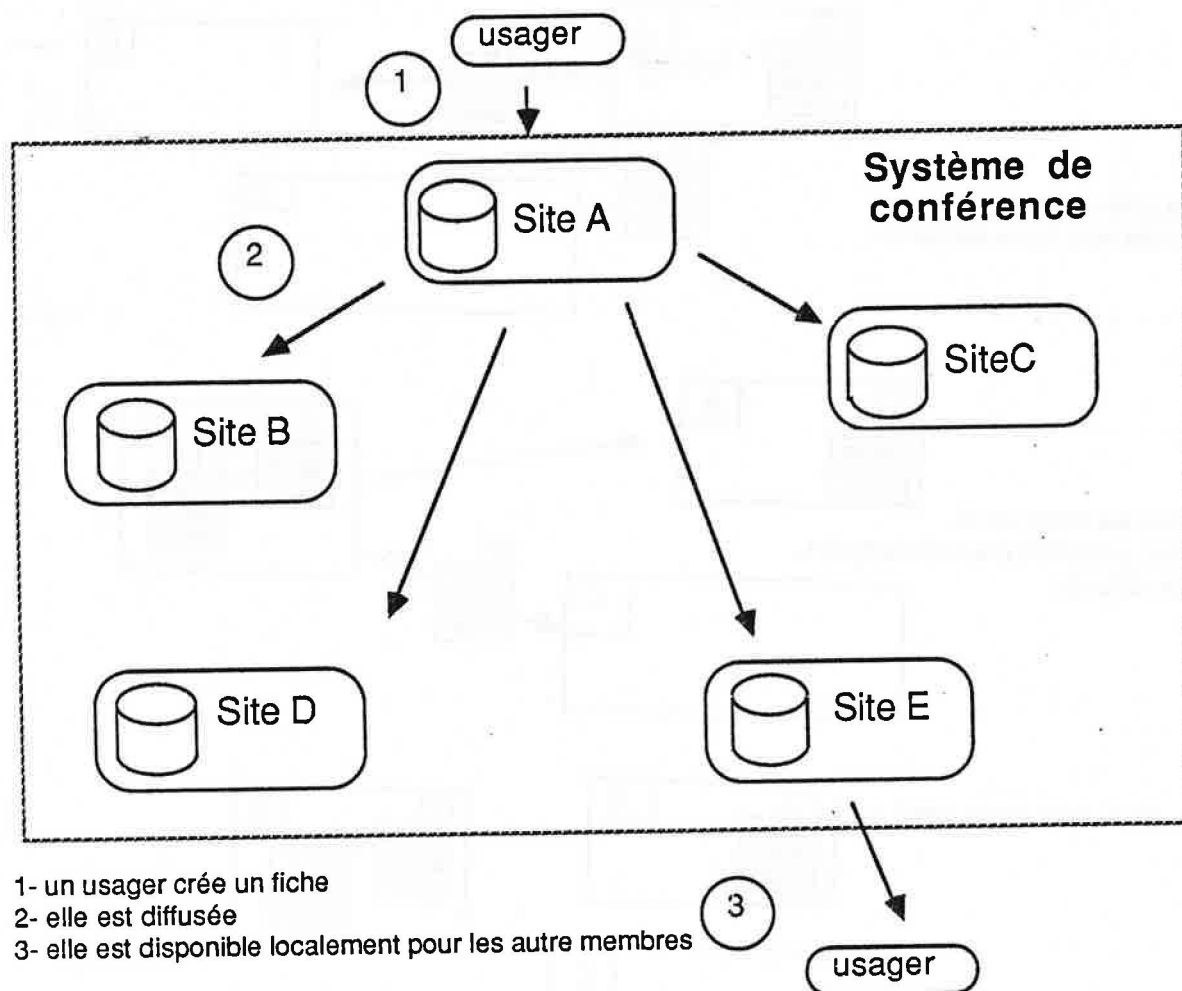
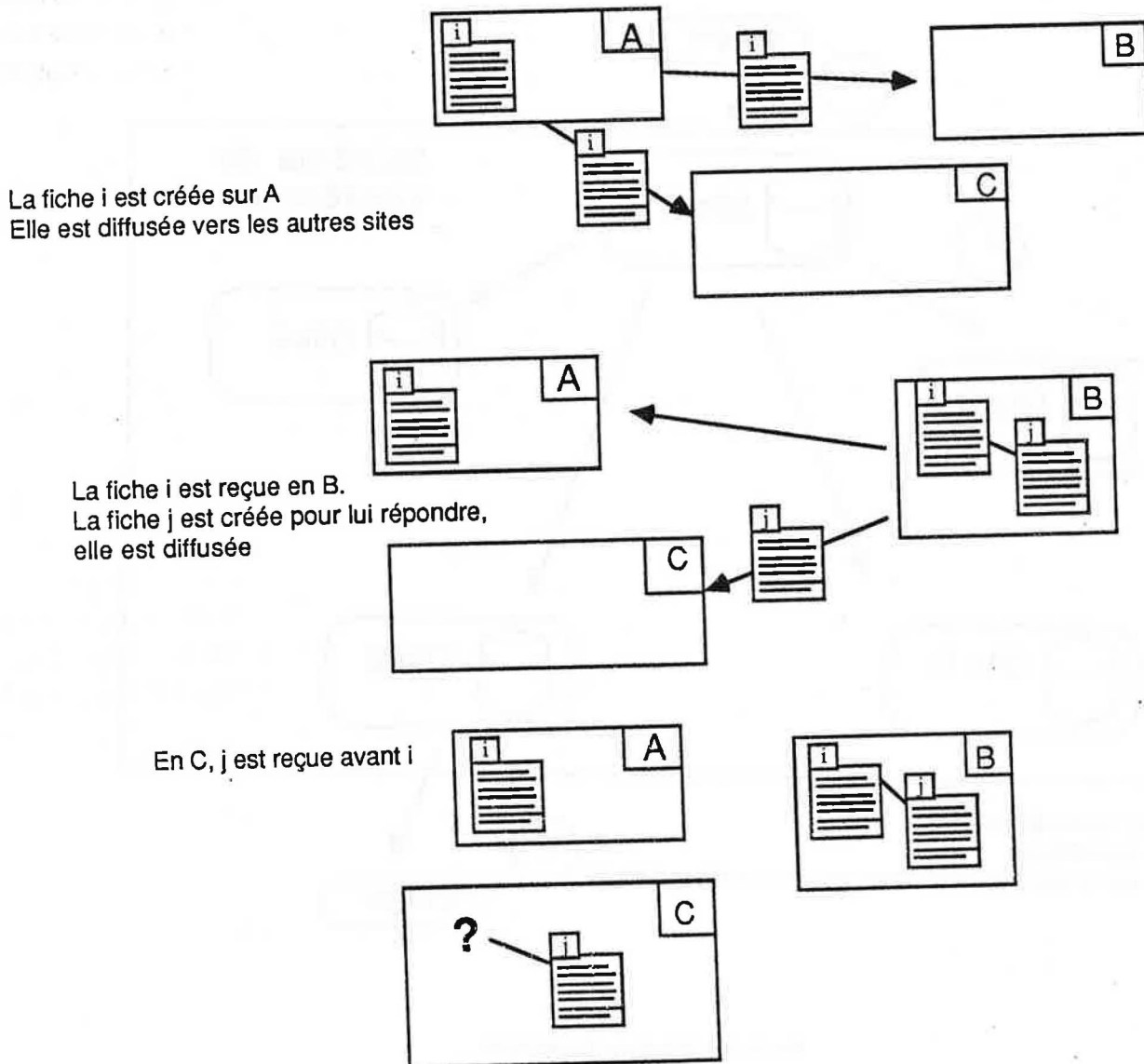


Fig III.3 : Diffusion d'une fiche

### 2.5.1 Synchronisation des messages

Toutefois, des problèmes de cohérence apparaissent entre les différentes versions de la base. Quel que soit le mode de transmission choisi, la diffusion d'un message vers l'ensemble des sites n'est pas instantanée. Aussi, la séquence des messages échangés dans le système risque d'être bouleversée. Il arrive que des informations soient reçues avant d'autres émises antérieurement.

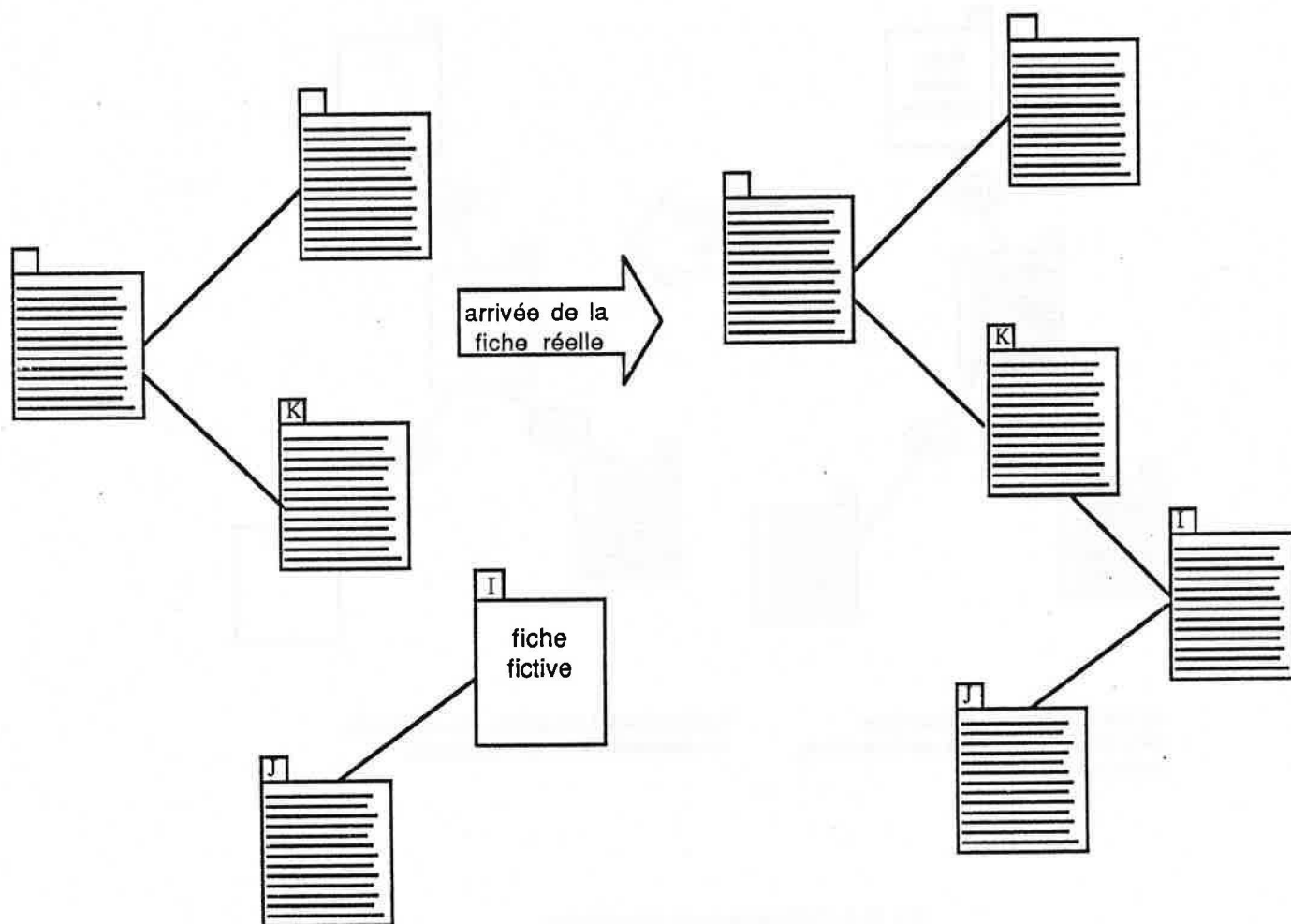
Considérons l'exemple suivant (Fig. III.4) avec trois sites A, B et C. Sur A un usager ajoute une fiche *i*, sa création est diffusée vers les autres sites. Elle est reçue en B où un utilisateur apporte un commentaire. La fiche qui la contient fait référence à *i*. Elle est alors envoyée vers A et C. Si *j* est reçue avant *i* sur C, cette dernière est inconnue lors de l'arrivée de *j*.



Cet exemple montre les incohérences possibles entre les versions. A moins de mettre en place un système complexe et fragile de confirmations entre les sites ou d'adopter une configuration fortement centralisée, il est impossible d'assurer un maintien parfait de la séquence des messages.

Il faut que CRMM s'adapte à ces incohérences passagères entre les versions. Une solution serait de retarder la prise en compte d'une fiche tant que toutes les références ne sont pas résolues localement. Il faudrait alors veiller aux risques d'interblocage entre les fiches en attente.

La solution retenue consiste à incorporer les nouvelles fiches dès leur réception. Lorsqu'elles font allusion à une fiche inconnue localement, une fiche vide dite **fiche fictive** est créée. Lorsqu'elle arrive, la fiche réelle la remplace en tenant compte des liens qu'elle peut mémoriser (Fig. III.5).



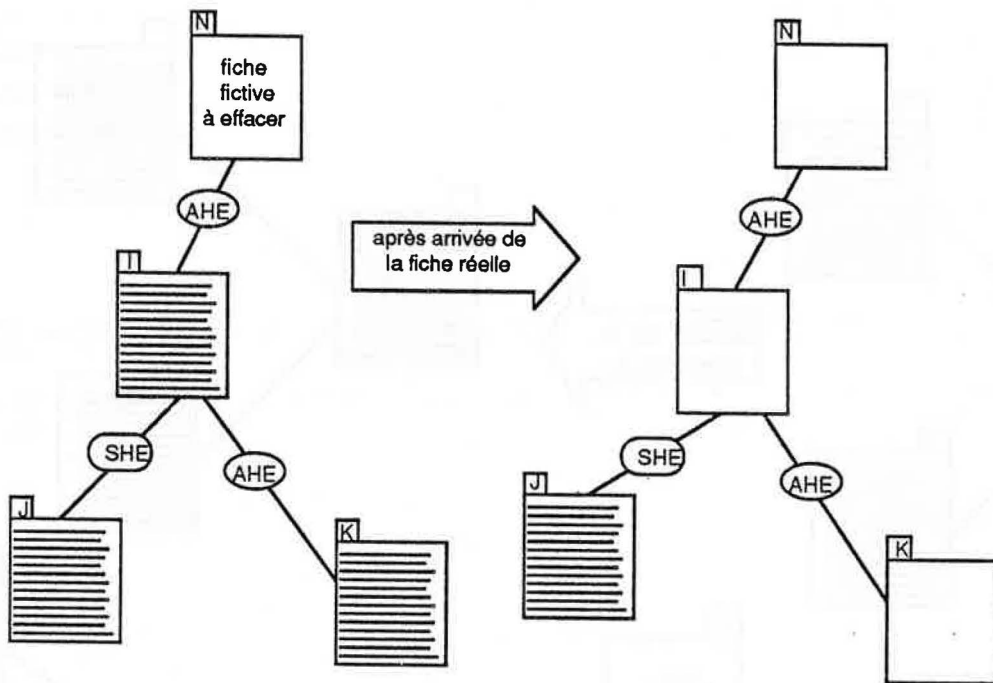
La fiche J arrive sur le site, elle est liée à une fiche I qui est inconnue. Une fiche fictive I est créée.

La fiche réelle I arrive sur le site. Elle remplace la fiche fictive. Elle est liée à la fiche K.

Fig III.5 : Fiche fictive



Une fiche fictive peut aussi mémoriser une action que la fiche réelle doit subir. Par exemple, supposons que le site reçoive une demande d'effacement de la fiche N qui est inconnue. Dans ce cas, une fiche fictive est créée et elle mémorise l'action que doit subir la fiche réelle. Lorsque celle-ci arrive, elle est effacée et sa descendance est soumise à l'héritage de l'effacement. (Fig. III.6). Si un ordre de destruction concerne une fiche fictive, le traitement n'est pas effectué, mais l'ordre est mémorisé. La fiche réelle sera détruite à son arrivée.



Le site reçoit l'ordre d'effacement de la fiche N qui est déjà une fiche fictive, l'ordre est mémorisé.

La fiche réelle I est arrivée, elle est effacée, sa descendance subit l'effacement.

Fig III.6 : Effacement de fiche fictive

Le mécanisme décrit précédemment ne prétend pas répondre à tous les problèmes de cohérence entre les versions de la base. Elles risquent de différer. Ceci montre les limites du produit CRMM, il ne s'agit pas de réaliser une base de données répartie sur différents sites. Ce type de produit pose des problèmes qui dépassent largement le cadre de cette étude. Pour autant, le produit décrit ici, garde tout son intérêt et bien que plus simple, rend de grands services aux utilisateurs.

De plus, même si une cohérence parfaite était théoriquement possible, elle ne serait pas obligatoirement souhaitable. En effet, accepter des bases incohérentes autorise beaucoup plus de souplesse. Par exemple, les administrateurs locaux modifient leur base sans diffuser les changements. Cela est utile pour répondre à des problèmes d'exploitation tels que l'espace mémoire utilisé. La destruction de fiches est réservée à l'administrateur. Si ce dernier est seulement administrateur local d'un site, la destruction ne sera pas diffusée, l'opération ne concerne que sa version de la base. En revanche, s'il s'agit de l'administrateur général, toutes les modifications sur base sont diffusées. Ceci exige que l'administrateur général fasse fonction d'administrateur local sur son site.

## 2.6 DESIGNATION

La répartition pose des problèmes pour désigner les usagers et les fiches de CRMM. Pour les résoudre, il est nécessaire que les sites aient un nom unique. Ce nom est utilisé pour nommer les fiches et les usagers.

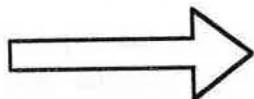
### 2.6.1 Désignation des fiches

Pour gérer les fiches non lues, chaque site ne doit pas avoir à mémoriser la liste exhaustive des fiches lues par chaque usager. Cela est évitable si les fiches sont numérotées dans l'ordre de leur arrivée. Ainsi pour usager donné, il suffit de mémoriser :

- le numéro le plus élevé tel que les précédents correspondent à des fiches lues,
- la liste des numéros de fiches lues, supérieurs au numéro précédent (Fig. III.7).

liste des fiches lues

1
2
3
4
5
6
9
12
14



ce qui est mémorisé

6
9
12
14

numéro maximum tel que  
les précédents correspondent  
à des fiches lues

numéros supérieurs à 6  
qui correspondent à des fiches lues

si le numéro le plus élevé est 15, les fiches non lues ont pour numéro 7, 8, 10, 11, 13, 15

Fig III.7 : Gestion des fiches lues

Comme l'ordre d'arrivée des fiches diffère d'un site à l'autre, chacun d'eux dispose d'une numérotation qu'il lui est propre. Donc, une fiche est connue sur un site par un **numéro local**.

Ce numéro local est insuffisant, en effet il faut disposer d'un identifiant valable sur tous les sites. Par exemple, lorsqu'une fiche est diffusée, les fiches auxquelles elle est liée doivent être désignées de façon universelle. De plus, lorsque les usagers citent une fiche, il faut qu'ils disposent d'une désignation compréhensible partout.

Pour cela, un **identifiant global** est défini. Il est formé par le nom du site d'origine de la fiche, et du numéro local que ce site lui a affecté à sa création.

### 2.6.2 La désignation des usagers.

De la même façon, les usagers sont connus par le nom du site d'appartenance et par leur nom local à ce site.

## 2.7 FORMAT DES FICHES

Après répartition de la conférence, le descripteur de fiches est modifié. Il comprend désormais un identifiant global et un numéro local. De plus, l'état de la fiche indique si une fiche est fictive et les éventuelles opérations que la fiche réelle doit subir à son arrivée.

- L'identifiant global.
- Le numéro local à chacun des sites de la conférence.
- Le nom de l'auteur.
- Le nom de l'éventuel éditeur.
- La date de prise en compte du message.
- La durée de validité de la fiche.
- L'objet du message.
- Les mots clés.
- Les liens sur d'autres fiches.
- L'état de la fiche, par exemple fiche fictive.
- Le corps du message.

## 2.8 RECEPTION DE MESSAGES ERRONES

Dans tout système, il faut prévoir un mécanisme de gestion d'erreurs. Cette nécessité existe pour les processus indépendants mais la coopération entre entités coopérantes accroît le problème. Un processus autonome peut se terminer et laisser le système dans un état cohérent. En revanche, un processus inclus dans une activité plus vaste, doit veiller à ne laisser aucune autre tâche en attente. Les choix dans CRMM s'appuient sur les considérations suivantes.

- Sur réception d'un message erroné, le site remet une indication à l'exploitant et tente de prévenir l'expéditeur. Toutefois, il s'en abstient s'il y a risque d'amplification. Par exemple, sur réception d'un message qui signale une erreur et qui est lui même erroné, le site ne renvoie rien. Ces renvois supposent que le message est en partie compréhensible.
- Chaque site vérifie que l'expéditeur est bien inscrit dans la liste de distribution. Aucun message d'origine inconnue n'est traité.
- La plupart des communications se font en mode diffusion. Une gestion des acquittements serait lourde et complexe. Aussi, les messages d'erreur qui sont reçus par un site sont simplement mis à la disposition des services d'exploitation de la machine.

Les communications ne se font pas en mode connecté, l'absence de réponse valide ne bloque pas la marche d'un site. Cette gestion rudimentaire des erreurs peut amener des différences entre sites : des fiches sont absentes ou

ne sont pas détruites ou effacées. Comme le système n'exige pas une cohérence forte entre les sites, il supporte ces différences. Pour remédier à de trop fortes différences, CRMM prévoit une opération de restauration de la base qui consiste à demander des fiches auprès du site de référence.

## 2.9 LES MESSAGES ENTRE SITES

Ce paragraphe décrit les messages que les sites sont amenés à s'échanger. Pour chacun, sont donnés les paramètres qu'ils contiennent.

### 2.9.1 Diffusion d'opérations sur la base

Ces messages sont diffusés à l'ensemble des sites de la conférence à partir du site où une modification s'est produite .

- Insertion de fiche :
  - la fiche à insérer.
- Effacement de fiche :
  - Identifiant global de la fiche concernée.
- Destruction de fiche :
  - Identifiant global de la fiche concernée.
- Ajout de lien (implicitement, il s'agit d'un lien neutre) :
  - Identifiants globaux des deux fiches concernées.
- Suppression de lien :
  - Identifiants globaux des deux fiches concernées.

### 2.9.2 Message de services

Ils sont envoyés par l'administrateur général.

- nom d'éditeur par défaut :
  - nom de l'utilisateur éditeur.
- liste de diffusion, avec pour chaque item :
  - nom du site,
  - droits accordables par l'administrateur.
- restauration de fiches :
  - fiches redistribuées

### 2.9.3 Imprimatur

- demande d'imprimatur :
  - fiche,
  - demande de retour de fiche,
  - nom de l'éditeur concerné.
  
- réponse à demande d'imprimatur :
  - nom de l'auteur,
  - nom de l'éditeur,
  - réponse,
  - commentaire de l'éditeur,
  - la fiche si son retour a été demandé.



### 2.9.4 Messages d'erreur

- fiche connue : demande d'insertion de fiche dont l'identifiant est déjà affecté.
- lien connu : demande d'ajout de lien entre deux fiches déjà liées.
- lien inconnu : demande de suppression de lien non existant.
- éditeur inconnu : la fiche est soumise à un usager inconnu du site
- éditeur incapable : la fiche est soumise à un usager qui n'est pas éditeur.

### 3 INTEGRATION DANS LA MESSAGERIE X400

Le système de conférences étudié implique des communications distantes entre les sites où sont gérées les différentes versions de la base des messages. L'outil choisi pour acheminer les informations respecte les recommandations X400. Il s'agit d'un service de messagerie.

Après avoir défini généralement la messagerie, cette section décrit celle qui est recommandée par le CCITT. Ensuite, nous verrons comment CRMM utilise les services de transfert de messages. Enfin, la dernière partie décrit la nouvelle application ainsi construite en respectant la présentation habituelle (protocole, primitive).

#### 3.1 LA MESSAGERIE

Les modes de transmission classiques comme le téléphone ou le télex offrent un service de point à point vers un seul destinataire. De plus, ils exigent que l'émetteur et le récepteur soient disponibles pendant la communication. Dans le cas du télex, le terminal doit être sans arrêt à l'écoute.

Le développement des boîtes aux lettres électroniques, bien adaptées aux communications de personne à personne dans un environnement bureautique, et les progrès dans la commutation de messages (utilisation du code ASCII, débit plus élevé, terminaux performants) ont permis un mode de communication plus élaboré, la messagerie électronique.

Les principales caractéristiques des boîtes aux lettres sont les suivantes.

- Elles sont accessibles par une adresse unique.
- Elles gardent une trace du message. Tant que celui-ci n'est pas détruit par le destinataire, il demeure dans la boîte.
- Elles présentent énormément d'intérêt pour les destinataires peu disponibles ou souvent absents, car elles n'exigent pas la présence simultanée des deux correspondants.

- Le destinataire doit consulter sa boîte aux lettres régulièrement, une consultation au moins quotidienne est nécessaire, sinon le courrier classique devient plus intéressant. Pour que l'interrogation de sa boîte devienne habituelle, le nombre de messages reçus doit être suffisamment important.
- Actuellement, le peu de messages échangés est un obstacle à la consultation régulière. La messagerie électronique n'a pas atteint le développement du télex. Il est encore peu courant de s'échanger ses adresses électroniques.
- Certaines facilités d'utilisation sont évidemment possibles. Par exemple, lors de la connexion au terminal, l'utilisateur est averti du nombre de messages non encore lus, ou encore, si un message est déposé dans la boîte alors que le destinataire est présent, ce dernier est prévenu par un signal lumineux ou sonore.
- Les boîtes offrent une meilleure confidentialité que le télex dont la réception automatique qui est un atout par ailleurs, est gênante dans ce cas.
- Contrairement au télex, les utilisateurs ont un accès rapproché à leur boîte. Grâce à la bureautique, de plus en plus d'utilisateurs ont un terminal disponible à proximité. Actuellement, la consultation se fait à distance, l'utilisateur se connecte à la machine où se trouve sa boîte pour lire son courrier. De gros efforts sont faits pour permettre un accès à partir d'un poste de travail, des protocoles sont en cours de définition pour une bonne coopération entre la machine où se trouve la boîte et la machine du lecteur [BRUN87].
- Le terminal utilisé n'est pas monopolisé par la communication. Il n'a pas à être sans arrêt à l'écoute. Il est utilisable pour d'autres services ou traitements locaux.
- Bien adapté à la communication de personne à personne, ce mode souffre de carences lorsque la communication a lieu entre entreprises ou autres institutions. Il demande des règles supplémentaires de la part des correspondants pour lire les messages, les écrire et les retransmettre si nécessaire.



### 3.1.1 Fonctions d'un système de messagerie

Parmi les services qu'offre une messagerie, on retrouve les possibilités du courrier traditionnel (urgence, avis de remise), des options imposées par les matériels utilisés (conversion) et des fonctions propres aux nouveaux outils de communication (remise différée, multideestination).

#### urgence du message

L'expéditeur choisit entre différents niveaux tels qu'urgent, non-urgent, ou normal. Ceci correspond à des délais de transmission plus ou moins courts. Certains systèmes acheminent les messages non urgents pendant les heures creuses, la nuit par exemple. Pour d'autres, le degré d'urgence donne le temps au bout duquel un message qui n'a pu être transmis est retourné à l'expéditeur. Ainsi, les messages urgents non transmis reviendront plus rapidement que les messages non-urgents.

#### remise à plusieurs destinataires

Le nombre des destinataires n'est pas nécessairement limité et la remise n'est pas obligatoirement simultanée. Il est possible de cacher à chacun d'entre eux la liste des autres destinataires. De plus, des destinataires suppléants peuvent être donnés par l'expéditeur.

#### conversion

Le codage des informations change, par exemple de l'ASCII vers le télex. Les procédures de transmission et le format des messages peuvent eux aussi évoluer, par exemple la mise en page varie pour les modes Télétex et Vidéotex.

#### remise différée

Un expéditeur a la possibilité de demander que ses messages ne soient pas remis avant la date qu'il indique.

#### remise ultérieure

De même, un usager peut vouloir que les messages qui lui sont destinés, soient mis en attente jusqu'à une certaine date.

#### essai

Le système permet à un utilisateur de tester si un message pourra être transmis. Ceci est utile pour vérifier la validité d'une adresse.

### édition

Un service de messagerie peut comprendre des possibilités d'édition pour préparer des messages. Ceci fait partie des facilités locales qui n'influent pas sur le protocole entre entités coopérantes. Toutefois, elles participent à la définition du produit.

### avis de remise

L'utilisateur du service de messagerie peut demander à être prévenu des échecs ou des réussites dans l'acheminement de ses courriers.

## 3.2 LA MESSAGERIE X400

La messagerie X400 est le résultat d'un travail du CCITT (Comité Consultatif International Télégraphique et Téléphonique) qui a repris des propositions de l'IFIP (International Federation for Information Processing). Elle définit un ensemble de normes permettant l'échange international de messages entre abonnés du service. Elle s'intègre dans le modèle de référence OSI [OSI84] qui décrit la communication entre systèmes hétérogènes. Les protocoles de la messagerie appartiennent à la septième couche de ce modèle, elle est nommée **couche application**. La messagerie X400 est décrite plus en détail dans la série de recommandations X400 du CCITT [CCITT84].

### 3.2.1 Description générale

Les communications suivent le mode enregistrement et retransmission (*store and forward*). Il n'y a pas de connexion entre les correspondants, le message est transféré de proche en proche depuis le système où se trouve l'expéditeur vers celui du destinataire.

Deux services ont été définis. D'abord, la Messagerie de Personne à Personne (IPM pour *InterPersonal Message*) rend le service de courrier électronique. Ensuite, le service de Transfert de Messages (MT pour *Message Transfert*) concerne le service général de transfert de messages. Il est utilisé par la messagerie interpersonnelle mais sert pour d'autres services tels que le transfert de fichiers, qui n'ont pas encore été normalisés.

## Les entités du modèle

Le modèle du système de messagerie (MHS pour *Message Handling System*) décrit le rôle et le fonctionnement des différentes entités fonctionnelles qui rendent les services de messagerie.

- L'**utilisateur** est un individu ou une application informatique utilisant le service de messagerie. Il est désigné comme **expéditeur** lorsqu'il envoie des messages vers les **destinataires**.
- L'**Agent Utilisateur (UA pour User Agent)** est l'entité qui sert à l'utilisateur pour préparer, émettre, recevoir et stocker les messages. Des classes regroupent les UA coopérants qui traitent le même type de message. Par exemple, les UA du courrier interpersonnel sont des IPMUA pour *InterPersonal Message User Agent*. Ils dialoguent avec le MTA pour *Message Transfert Agent*. (Fig. III.8).

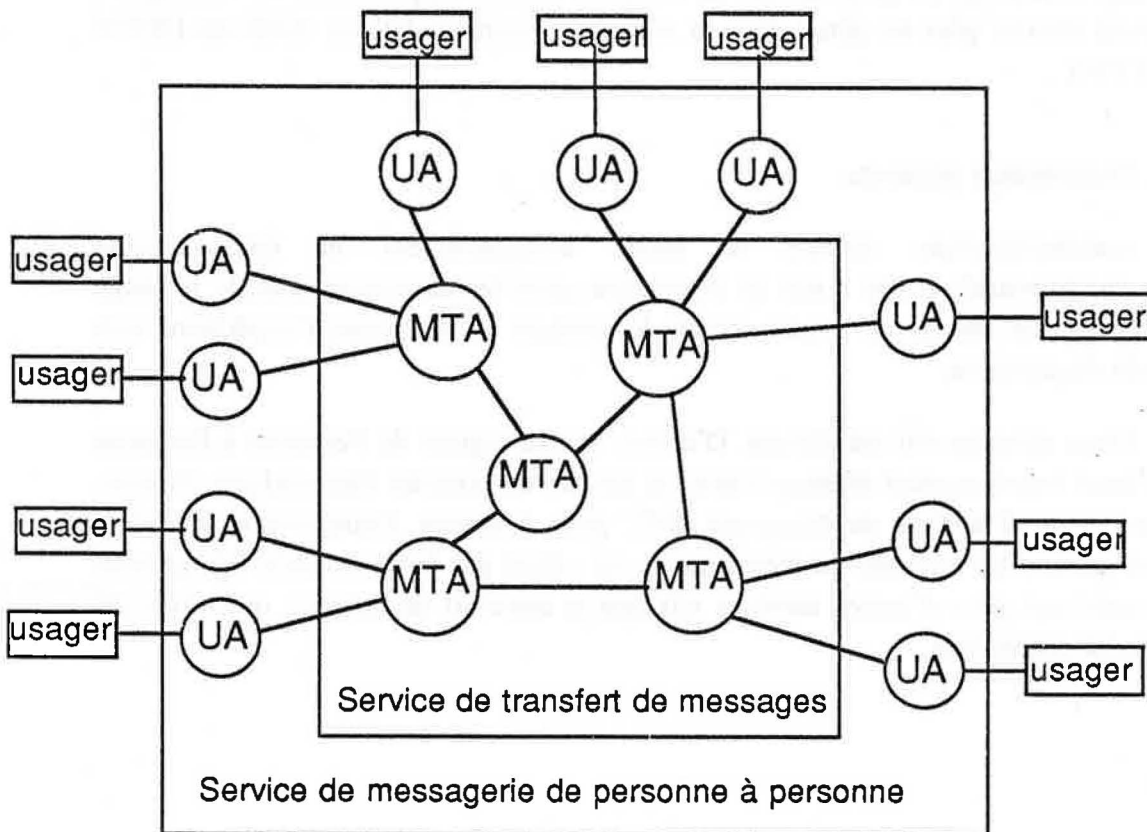


Fig III.8 : Modèle MHS

Un message est constitué d'une **enveloppe** et d'un **contenu**. L'enveloppe contient différentes informations parmi lesquelles on trouve les noms de l'expéditeur et des destinataires, la date d'émission, une demande d'accusé de réception. Elle sert au routage entre MTA. Le contenu du message peut être du texte, de la voix ou des images. Un message a un ou plusieurs destinataires.

### adressage

Un ensemble constitué par au moins un MTA et zéro, un ou plusieurs UA appartenant à un même organisme constitue un **Domaine de Gestion** (MD pour *Management Domain*). Un MD géré par un organisme public (PTT ou équivalent) est appelé **Domaine de Gestion Administratif** (ADMD pour *ADministrative Management Domain*). Un MD géré par un organisme privé est appelé **Domaine de Gestion Privé** (PRMD pour *PRivate Management Domain*). Les recommandations CCITT (X400) décrivent le fonctionnement des messageries publiques (ADMD) (Fig. III.9).

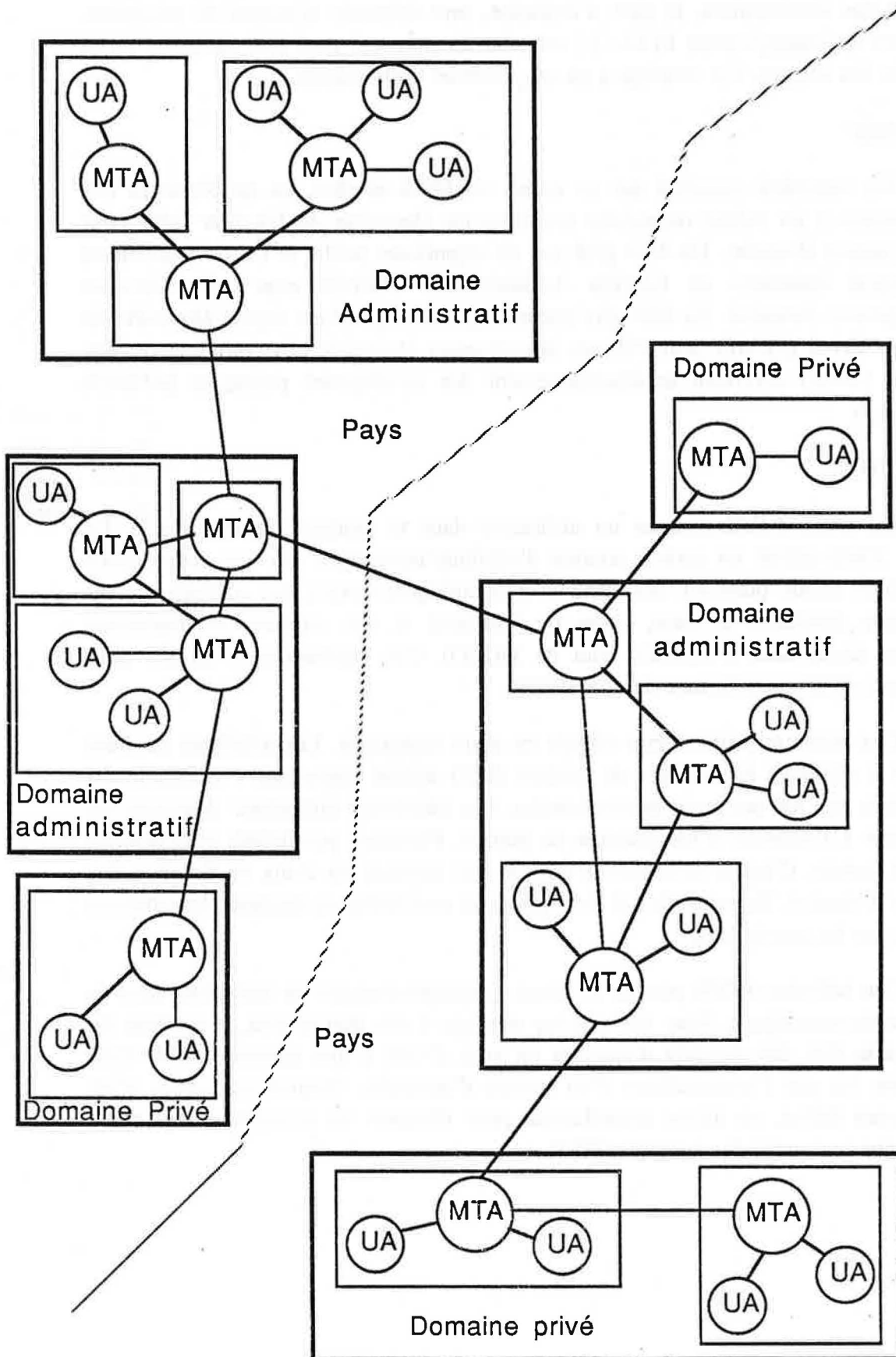
### Nom d'O/R

Un **nom d'O/R** désigne un utilisateur dans le système de messagerie. La norme X400 définit un certain nombre d'attributs normalisés tels que des attributs individuels (nom, prénom), des attributs géographiques (pays), des attributs organisationnels (sociétés, division, poste ou fonction) et des attributs architecturaux (adresse télex, nom d'ADMD, nom de PRMD). Ces attributs sont utilisés pour construire différentes formes de nom d'O/R.

Ces attributs peuvent être classés en deux catégories. Les **attributs de base** servent à désigner le domaine de gestion (MD) auquel appartient l'utilisateur. Ils sont fixés par les instances internationales. Les **attributs utilisateur** désignent un utilisateur à l'intérieur d'un domaine de gestion. Plusieurs possibilités sont offertes dans la norme. C'est le domaine de gestion qui effectue un choix en fonction des UA qu'il dessert. En revanche, il doit laisser la possibilité de désigner les attributs choisis par les autres MD.

Une **adresse d'O/R** permet de situer géographiquement un utilisateur dans le système de messagerie. Pour remettre un message à son destinataire, le système de messagerie doit être capable d'associer un nom d'O/R à une adresse d'O/R. Cela peut être fait par l'intermédiaire d'un service d'annuaire. Comme ce dernier n'est pas encore défini, on utilise actuellement pour désigner les utilisateurs des noms d'O/R qui sont aussi des adresses d'O/R.

Fig III.9 : Domaines d'adressage X400



## Configuration physique

Un UA et un MTA peuvent appartenir à un même système (Fig. III.10). Ils sont dits corésidents. Les usagers accèdent au service à partir d'un simple terminal (TTY ou vidéotex). Un UA est dit autonome s'il se trouve sur un système distinct (terminal intelligent). Un système peut contenir un MTA et aucun UA. On parle alors de MTA autonome.

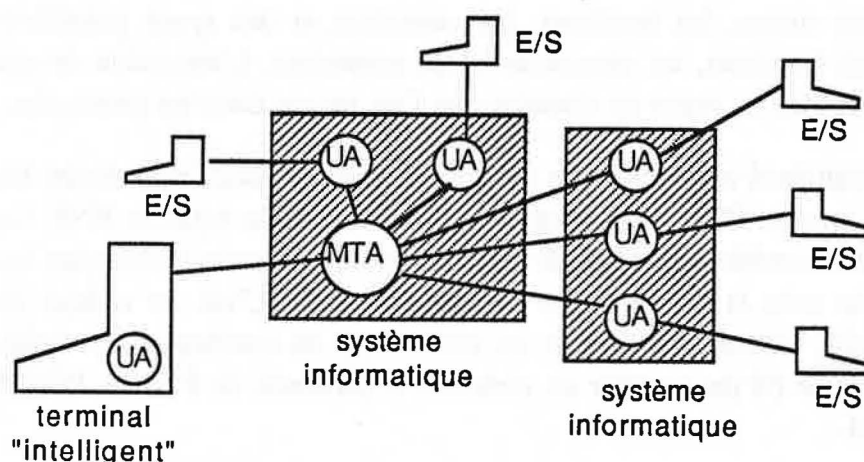


Fig III.10 : Configurations physiques

### 3.2.2 Représentation du modèle MHS

Les fonctions de messagerie sont réparties en deux couches dans la couche Application du modèle de référence OSI (Fig. III.11). La **couche Agent Utilisateur (Couche UAL)** contient les fonctions d'UA. Elle est constituée d'un ensemble d'entités d'UA (UAE) qui dialoguent à l'aide d'un protocole de nom générique *Pc*. Chaque classe d'UA coopérants suit un protocole spécifique. Par exemple, le *P2* est la version utilisée par les IPMUA. (Fig. III.12)

La **couche transfert de Messages (MTL pour *Message Transfert Layer*)** fournit les services de Transfert de Messages. Elle est constituée de MTA et d'entités de Dépôt et de Remise (SDE). Ces dernières gèrent un protocole d'accès à un MTA pour les UA autonomes. Les MTAE dialoguent à l'aide du protocole *P1*. Les SDE utilisent le protocole *P3* pour communiquer avec les MTAE.

### 3.2.3 Syntaxe X409

La syntaxe X409 (norme ASN1 à l'ISO) est le moyen utilisé dans les applications CCITT pour décrire et coder les différentes informations constituant les protocoles de communications.

Chaque information a un type et une valeur. Il existe des types prédéfinis primitifs tels que les entiers, les booléens, les caractères et des types prédéfinis constructeurs tels que les choix, les séquences et les ensembles. L'ensemble de ces types permet de construire les types de données que l'on trouve dans les protocoles.

La **notation standard** est l'ensemble des conventions utilisées pour écrire les protocoles dans les normes. C'est un langage décrit à l'aide de la notation BNF. La **représentation** est l'ensemble des règles de codage employées pour transmettre les données de protocole sous la forme d'une séquence d'octets. C'est un codage de forme Type, Longueur, Valeur (TLV). Elle est utilisée par de nombreux protocoles d'applications comme le P2 du courrier de personne à personne, le P10 de TFMM ou le P30 de CRMM.

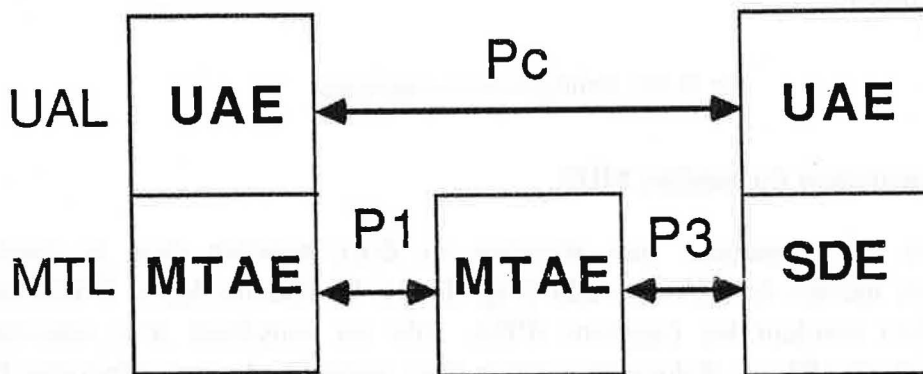


Fig III.11 : Description structurée en couches du système de messagerie

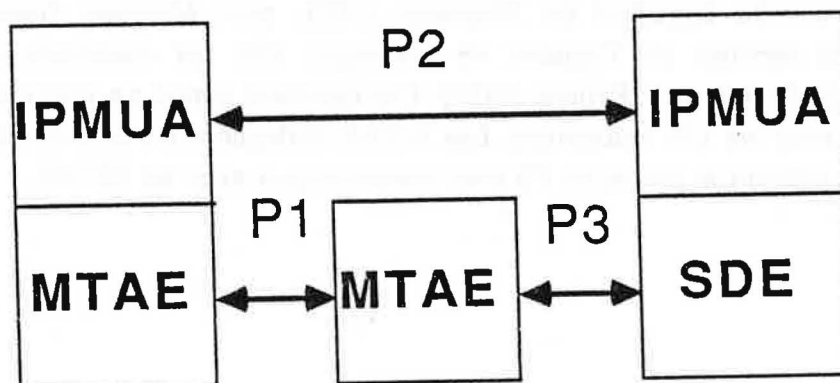


Fig III.12 : Représentation en couches du courrier de personne à personne

### 3.2.4 La messagerie COSAC

La messagerie Cosac (COmmunications SANS Connexion) réalisée au CNET est celle utilisée dans le projet CRMM. Elle respecte les avis du CCITT. Elle est composée d'un ensemble de MTA qui suivent le protocole P1, elle permet d'interconnecter des messageries privées non normalisées comme celles des systèmes Unix et Multics (Fig. III.13).

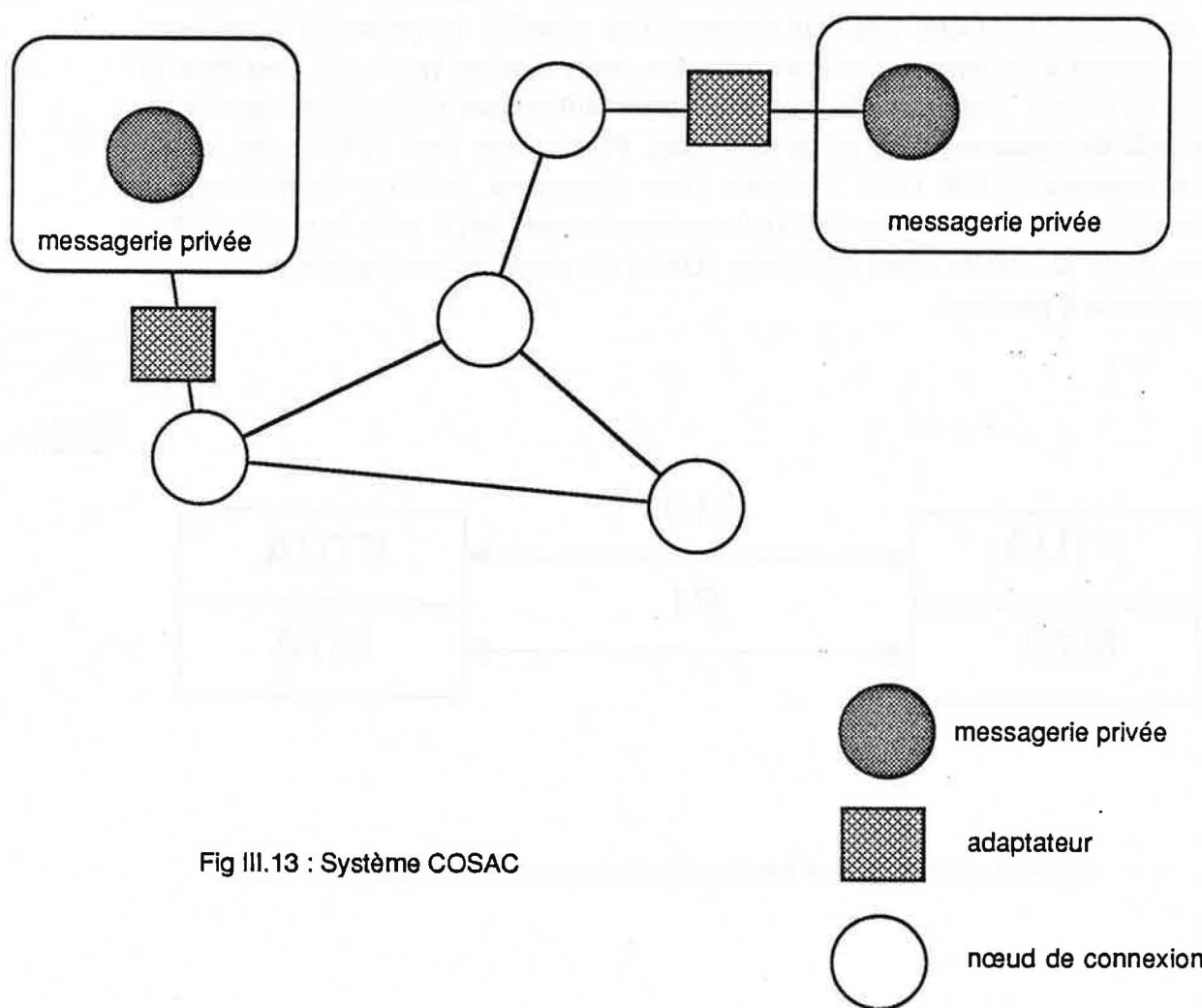


Fig III.13 : Système COSAC



### 3.3 CRMM ET LA MESSAGERIE

Actuellement, le courrier de personne à personne est la seule application utilisant la couche transfert de message à avoir été normalisée. D'autres outils de communication sont en projet, en cours de spécification ou en train d'être réalisés. Par exemple, TFMM (Transfert de Fichier en Mode Messagerie) utilise la messagerie pour envoyer ou prendre des fiches sur une machine éloignée. En plus des octets contenus dans le fichier, cet outil prend en compte la nature du fichier, ce qui permet des échanges entre systèmes d'exploitation hétérogènes. Ce produit est en phase d'industrialisation. STMM (Soumission de Travaux en Mode Messagerie) permet de demander l'exécution de tâches sur un système donné à partir d'une machine distante.

Il faut avoir en tête que le système de transfert de messages (MTS) achemine les enveloppes quel que soit leur contenu. Les courriers de personne à personne appartiennent à un type de contenu particulier, mais d'autres types sont possibles. Il suffit de définir une nouvelle classe d'agents utilisateurs (UA) et de donner le protocole de communication qu'ils respectent. Par exemple dans TFMM, une classe d'UA nommés FTUA (*File Transfert User Agent*) est créée et leurs échanges suivent le protocole P10 (Fig. III.14). Le protocole entre MTA reste le protocole P1. Donc, seule la couche agent utilisateur (UAL) est modifiée par rapport au courrier de personne à personne.

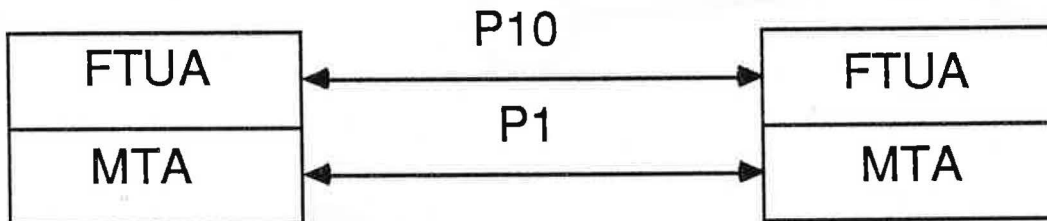


Fig III.14 : Description en couches d'un autre service de messagerie : TFMM

### 3.3.1 Conférences et courrier interpersonnel

Avant de définir une nouvelle classe d'UA pour le système de conférence, il faut se demander si les conférences ne sont pas une simple extension du courrier de personne à personne. Ceci permettrait d'utiliser les IPMUA pour rendre le service de conférence. En effet, ce dernier peut être vu comme un échange de courriers entre les membres de la conférence. Selon cette approche, chaque fois qu'un usager ajoute une fiche, un message la contenant est envoyé à tous les membres. Ceci s'assimile à une liste de diffusion par conférence comportant tous ses usagers. L'ajout ou le retrait d'utilisateurs revient à modifier cette liste (Fig. III.15).

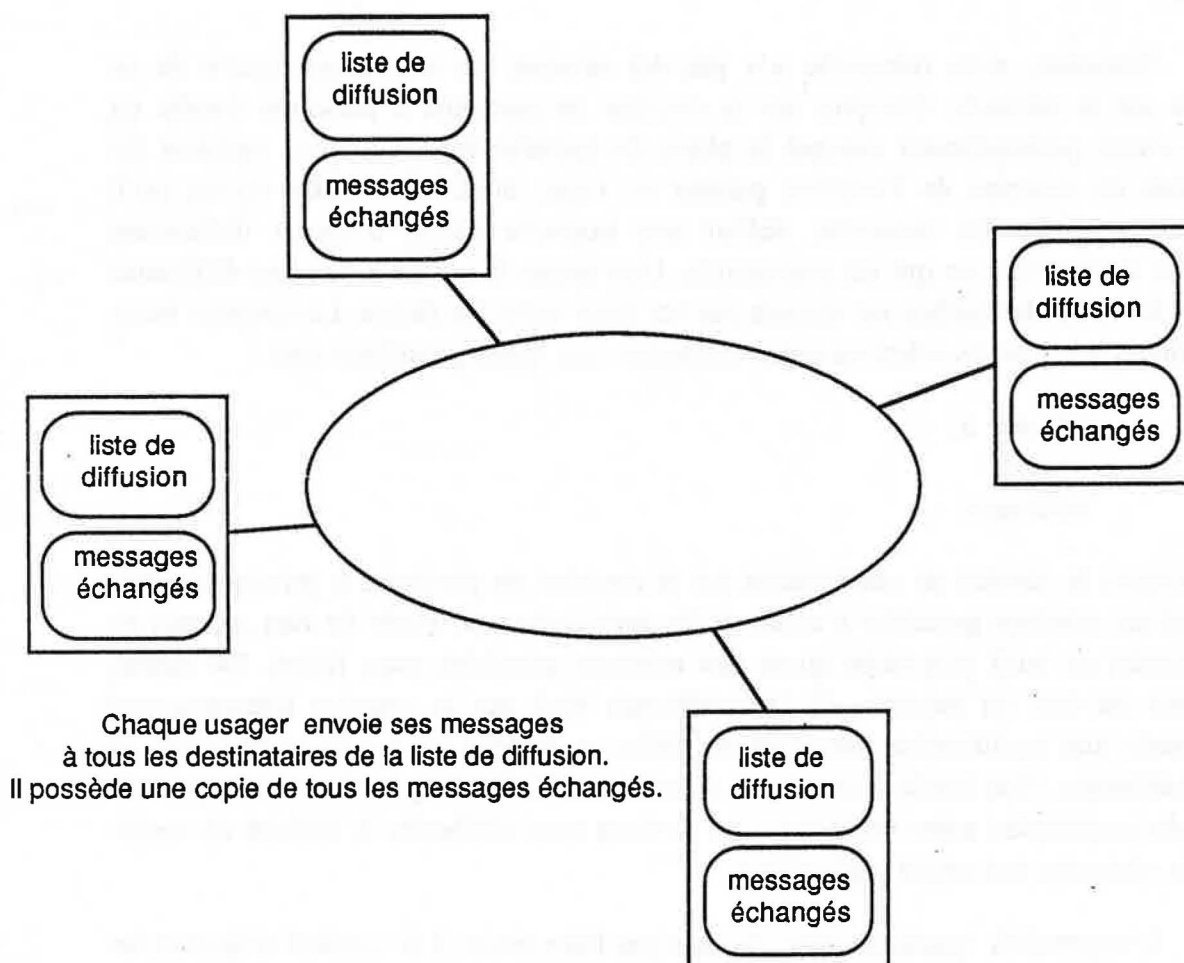


Fig.III.15 Système de conférences basé sur le courrier interpersonnel

La version actuelle du courrier de personne à personne ne permet pas une bonne gestion des listes de diffusion. Cependant ceci est amené à changer. Les actuels documents issus des travaux de normalisation datent de 1984. Depuis, des études autres que CRMM et des réalisations conformes aux propositions du CCITT ont vu le jour. Elles ont mis en évidence des améliorations souhaitables de ces premières recommandations. D'ailleurs, ces suggestions vont déboucher en 1988 sur une nouvelle version. Les modifications portent sur la couche transfert de messages (MTL) et sur la couche agent utilisateur (UAL), elles abordent notamment la gestion des listes de diffusion.

Une approche possible pour CRMM serait donc d'étendre les possibilités du courrier de personne à personne pour lui permettre de remplir la fonction de conférence.

Toutefois, cette démarche n'a pas été retenue. La raison essentielle de ce choix est la suivante. Compter sur le courrier de personne à personne étendu ou non, aurait profondément marqué la phase de spécifications. Celles-ci auraient été établies en fonction de l'existant présent ou futur, donc en fonction de ce qu'il rendait possible. En revanche, définir une nouvelle classe d'agents utilisateurs permet de spécifier ce qui est souhaitable. Une bonne illustration de cette différence entre les deux démarches est donnée par les liens entre les fiches. Le courrier interpersonnel autorise les relations entre messages. Les types prédéfinis sont :

- réponse à,
- annule,
- référence.

Construire le service de conférences sur le courrier de personne à personne, aurait limité les relations possibles à celles citées précédemment. Partir de rien, permet de construire un outil plus riche quant aux relations possibles entre fiches. De même, ajouter ou ôter un membre de la conférence basé sur le courrier interpersonnel demande une modification des listes de diffusion de tous les autres intervenants de la conférence. Plus fondamentalement, le service CRMM ne peut être rendu par une simple coopération entre individus. Les services sont différents. L'activité de conférence nécessite une entité particulière.

L'expression "partir de rien" ne doit pas faire croire à la volonté d'ignorer les travaux effectués pour le courrier de personne à personne. Bien au contraire, la plupart des participants au groupe de travail CRMM suivent ou même participent à ces travaux. Les options retenues dans notre système ne s'opposent pas à celles de la messagerie interpersonnelle. Elles tirent profit de l'expérience de ce service.

### 3.3.2 Description des entités

CRMM est donc un nouveau service de communications en mode messagerie telle qu'elle est décrite dans les recommandations X400. Il met en œuvre une nouvelle classe d'UA et repose sur un protocole spécifique P30.

Les UA intervenant dans cette nouvelle application de messagerie sont nommés **BBGA** : *Bulletin Board Group Agent* soit Agent de Groupe pour les Conférences. L'expression "Bulletin Board" capte la nature du service créé. Alors que les UA du courrier interpersonnel (IPMUA) représentent un usager (User Agent), les UA des conférences délèguent un groupe d'utilisateurs (Group Agent). Il existe un BBGA par site coopérant, sur chacun d'eux est abonné un ensemble d'utilisateurs. La description en couches est la suivante (Fig. III.16):

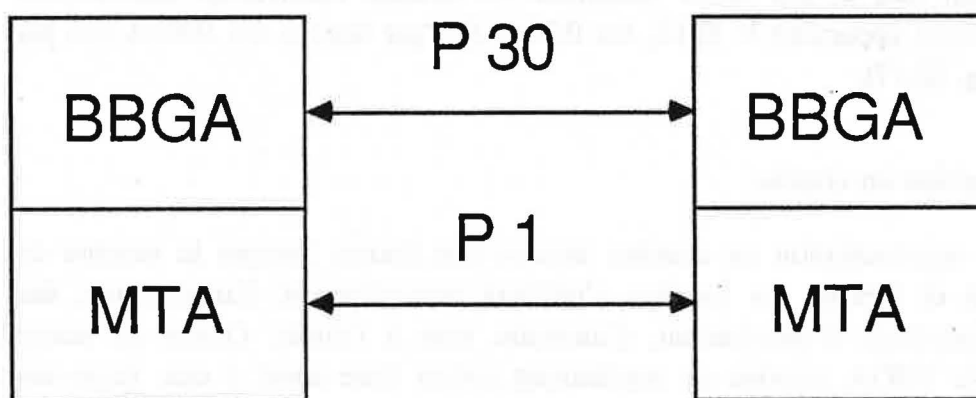


Fig III.16 : Description en couches de CRMM

Cette notion de groupe semble apporter une évolution par rapport à ce qui décrit dans les avis du CCITT. Toutefois, elle respecte le paragraphe 2.2.2.3 de la recommandation X400 où il est écrit :

*L'Agent Utilisateur utilise le service de MT (Transfert de Messages) fourni par le MTS (Système de Transfert de Messages). Un Agent Utilisateur est un ensemble de processus d'application informatiques qui comprennent, au minimum, les fonctions nécessaires à l'interaction avec le MTS, en utilisant les procédures de dépôt et de remise définies dans la Recommandation X.411<sup>1</sup>.*

1 X.411 : cette recommandation porte sur les principes généraux de la couche transfert de messages, les services de cette couche, le protocole de transfert et le protocole de dépôt et de remise.

Les UA de la classe définie dans CRMM respectent bien cette définition. En effet, ils communiquent par messages qu'ils déposent à l'intention de l'agent de transfert de messages (MTA) et qui leur sont remis par ce même MTA.

Ce qui change par rapport au courrier de personne à personne, c'est le fait que l'utilisateur final n'est plus en relation directe avec le BBGA qui est l'agent utilisateur du service MTS. Un membre d'une conférence accède aux services rendus par le BBGA à travers son BBUA, pour *Bulletin Board User Agent* (Agent Utilisateur de Conférence). Le terme Agent Utilisateur signifie que cette entité représente l'utilisateur final auprès du service de messagerie. Le BBUA ne se contente pas de transmettre requêtes et réponses entre l'utilisateur final et le BBGA, il offre des services qui lui sont propres. Par exemple, c'est lui qui prend en charge la gestion des fiches non lues et des fiches marquées. Le modèle fonctionnel obtenu pour CRMM fait donc apparaître le MTS, les BBGA (un par site) et les BBUA (un par membre) (Fig. III.17).

### 3.3.3 Description en couche

Le mode de représentation en couches montre des limites lorsque le nombre de services croît et lorsque ces services s'utilisent mutuellement. Par exemple, des services de stockage d'information, d'annuaire sont à l'étude. Quand ils seront disponibles, le BBGA (service de conférence) pourra faire appel à eux. Entre ces services coopérants, aucun niveau ne se distingue. A certains instants, un service est client d'un autre service, la relation peut être inversée quelques temps plus tard. Il apparaît entre les différentes entités des relations client-serveur.

### 3.3.4 Désignation

L'étude de la répartition a débouché sur une désignation des fiches et des membres à partir du nom des sites, donc des BBGA. Les BBGA en tant qu'utilisateurs du système de transfert de messages, sont connus par un nom d'O/R. Dans CRMM, la forme 1 variante 1 a été retenue comme seule valide. Chaque nom d'O/R est composé de :

- un nom de pays,
- un nom de domaine administratif,
- un nom de domaine privé,
- un nom de personne,
- un nom d'organisation,

- des noms d'unités organisationnelles,
- des attributs définis par un domaine.

Les deux premières informations sont obligatoires ainsi qu'un attribut au moins parmi le nom de domaine privé, le nom de personne, le nom d'organisation et les noms d'unités organisationnelles.

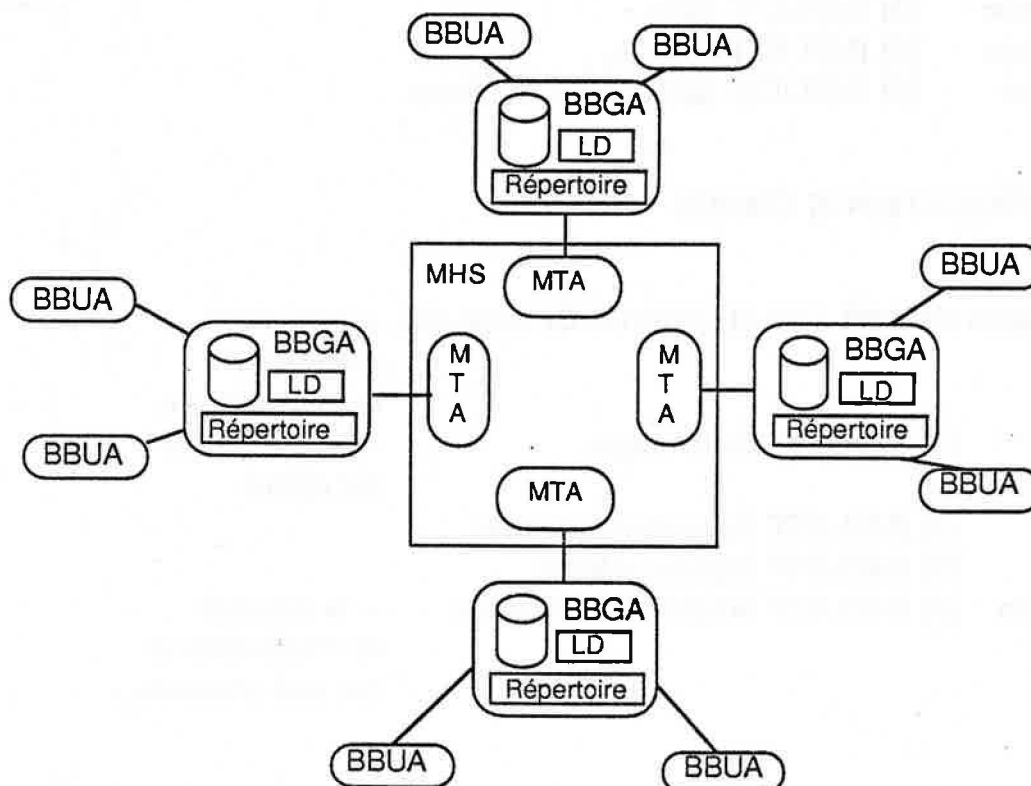
Le nom d'O/R est utilisé pour la communication entre sites. Pour la désignation des usagers et des fiches, le site est connu par un alias. Par exemple, le BBGA de la conférence Unix du site de l'Ecole des Mines aura comme nom d'O/R : */Fr/Ptt/Aristote/EMSE/ConfUNIX* et comme alias *EMSE*. L'alias doit être unique par conférence.

Un BBUA est connu par le BBGA auquel il est abonné et par un nom local. La désignation globale d'un BBUA ou de l'utilisateur qu'il représente est obtenue par concaténation de l'alias du BBGA d'attache et de son nom local.

La désignation globale d'une fiche est composée :

- de l'alias du BBGA d'origine de la fiche,
- du numéro local que ce BBGA lui avait affecté.

Par exemple, un écrivain abonné au BBGA A d'alias *EMSE* ajoute une fiche. Celle-ci est la cent troisième de la base du site A. Son numéro local sur A est 103, son identifiant universel est *EMSE:103*.



III : Répartition de la conférence

### 3.4 LE PROTOCOLE

P30 ::= [APPLICATION 0] CHOICE

```
{
opération-sur-base      [0] IMPLICIT Base,
message-de-service     [1] IMPLICIT Service,
message-erreur        [2] IMPLICIT Problème
}
```

Base ::= [APPLICATION 1] CHOICE

```
{
détruire-fiche   [0] IMPLICIT Id-fiche,
détruire-lien   [1] IMPLICIT SEQUENCE

    {
    première-fiche Id-fiche,
    seconde-fiche Id-fiche
    },
ajouter-lien    [2] IMPLICIT SEQUENCE

    {
    première-fiche Id-fiche,
    seconde-fiche Id-fiche
    },
insérer-fiche   [3] IMPLICIT Fiche,
effacer-fiche   [4] IMPLICIT Id-fiche,
redistribuer    [5] IMPLICIT SEQUENCE OF Fiche
}
```

Service ::= [APPLICATION 2] CHOICE

```
{
liste-diffusion [0] IMPLICIT SEQUENCE OF Liste-Site, -- nom d'O/R
                                                         des BBGA
                                                         de la conférence.
éditeur         [1] IMPLICIT Nom-Usager,           -- nom d'éditeur
                                                         par défaut.
                                                         [2] IMPLICIT Demande-Imprimatur,
                                                         [3] IMPLICIT Réponse-Editeur,
restauration    [4] IMPLICIT SEQUENCE OF Fiche    -- la demande
                                                         de restauration se
                                                         fait hors-protocole.
}
```

Problème ::= [APPLICATION 3] IMPLICIT SEQUENCE

```

{
raison-erreur          [0] IMPLICIT INTEGER

    {
    fiche-déjà-connue(0),
    lien-inconnu(1),
    lien-déjà-existant(2),
    réponse-éditeur-à-auteur-inconnu(3)    -- d'autres valeurs pourront
                                                être définies ultérieurement
    },
message-retourné      [1] CHOICE

    {
    [0] Base,
    [1] Service
    }
}

```

Liste-Site ::= SEQUENCE

```

{
nom-BBGA P1.NOM-OR,
écrivain BOOLEAN,      -- indique si le BBGA autorise les écrivains.
éditeur BOOLEAN        -- indique si le BBGA autorise les éditeurs.
}

```

Demande-imprimatur ::= [APPLICATION 4] IMPLICIT SEQUENCE

```

{
retour-fiche  [0] IMPLICIT BOOLEAN, -- vrai si le retour de la
                                                fiche avec la réponse,
                                                est souhaité par l'auteur.

nom-d-éditeur [1] IMPLICIT Nom-Usager,
nom-d-auteur  [2] IMPLICIT Nom-Usager,
fiche         [3] IMPLICIT Fiche
}

```

Réponse-Editeur ::= [APPLICATION 5] IMPLICIT SEQUENCE

```

{
nom-auteur          [0] IMPLICIT Nom-Usager,
commentaire         [1] IMPLICIT Chaîne-IA5,
nom-d-éditeur       [2] IMPLICIT Nom-Usager,
}

```



```

réponse          [3] IMPLICIT INTEGER

    {
    accordé(0),
    refusé(1),
    éditeur-inconnu(2),
    éditeur-incapable(3),
    éditeur-ne-répond-pas(4)
    },
[4] IMPLICIT Fiche OPTIONAL          -- si retour-fiche TRUE
}

```

Nom-Usager ::= [APPLICATION 6] IMPLICIT SEQUENCE

```

{
nom-du-BBGA      [0] IMPLICIT P1.NOM-OR,
nom-local       [1] IMPLICIT Chaîne-IA5
}

```

Fiche ::= [APPLICATION 11] SEQUENCE

```

{
Descripteur,
corps SEQUENCE OF Chaîne-IA5
}

```

Descripteur ::= [APPLICATION 10] IMPLICIT SEQUENCE

```

{
identifiant      [0] IMPLICIT Id-fiche OPTIONAL,
                 [1] IMPLICIT SEQUENCE OF Lien OPTIONAL,
auteur          [2] IMPLICIT Nom-Usager,
éditeur
                 [3] IMPLICIT Nom-Usager OPTIONAL,
date-de-crétion [6] IMPLICIT Heure-Généralisée OPTIONAL,
date-de-péremption [7] IMPLICIT Heure-Généralisée OPTIONAL,
objet           [8] IMPLICIT Chaîne-IA5,
mots-clés       [9] IMPLICIT SEQUENCE OF Chaîne-IA5
}

```

Lien ::= [APPLICATION 7] IMPLICIT SEQUENCE

```

{
type            [0] IMPLICIT INTEGER
}

```

```

    {
      séquentiel-avec-effacement(0),
      séquentiel-sans-effacement(1),
      hiérarchique(2),
      neutre(3)
    },
    valeur      [1] IMPLICIT INTEGER,      -- définie lors de l'initialisation
    fiche-liée  [2] IMPLICIT Id-fiche
  }

```

Id-fiche ::= [APPLICATION 9] IMPLICIT SEQUENCE

```

  {
    nom-du-BBGA-d-introduction  [0] IMPLICIT P1.NOM-OR,
    compteur-du-BBGA           [1] IMPLICIT INTEGER
  }

```

### 3.5 PRIMITIVES DE BBGA OFFERTES AUX BBUA

Le service de BBGA est décrit comme un ensemble de primitives de service offertes à l'entité BBUA. Seule la liste des services est donnée dans ce paragraphe. Les primitives sont détaillées en annexe.

- Sélection des Fiches.
- Lecture des Fiches.
- Consultation de la liste de BBUA.
- Consultation de la liste de diffusion.
- Destruction locale d'une fiche.
- Modification du nom du BBGA de référence.
- Modification de la liste des BBUA.
- Imprimatur
- Insertion de fiche.
- Effacement d'une fiche.
- Modification de la liste de diffusion
- Destruction d'une fiche.

- Destruction de lien.
- Ajout d'un lien neutre
- Changement de l'éditeur par défaut.
- Changement de la date de péremption.
- Redistribution d'une fiche.
- Restauration de la base.

### **3.6 PRIMITIVES DE BBUA OFFERTES A L'USAGER**

- Sélection de fiches
- Lecture des fiches.
- Consultation de la liste des usagers.
- Consultation de la liste de diffusion.
- Modification du nom du BBGA de référence.
- Modification de la liste des BBUA.
- Imprimatur
- Insertion de fiche
- Effacement d'une fiche.
- Modification de la liste de diffusion
- Destruction d'une fiche.
- Destruction de lien.
- Ajout d'un lien neutre
- Marquage de fiche.
- Restauration de fiche
- Redistribution d'une fiche.
- Modification de la date de péremption

- Modification du nom d'éditeur

### 3.7 QUELQUES SCENARII

Pour illustrer le fonctionnement du système, voici détaillés quelque exemples d'utilisation.

#### 3.7.1 Consultation de la base

La consultation est une opération purement locale, elle ne met pas en œuvre le système de transfert de messages.

L'utilisateur qui désire lire une fiche dans la base, adresse à son BBUA une demande de lecture en désignant la fiche voulue. Il peut le faire explicitement par la marque, le numéro local ou l'identifiant global, ou implicitement en demandant la fiche suivante non lue, la fiche suivante dans la sélection. Le BBUA transforme les identifications implicites en numéros locaux et transmet cette requête au BBGA. Si la réponse du BBGA est positive, elle inclut la fiche désirée. Une réponse négative est accompagnée de la raison du refus, par exemple "Fiche inconnue".

#### 3.7.2 Ajout d'une fiche par un écrivain

L'auteur prépare la fiche grâce à son BBUA, elle est soumise au BBGA (Fig. III.18). Ce dernier vérifie les droits du demandeur, à savoir s'il est bien écrivain. De plus, les fiches liées à la fiche proposée doivent être connues localement. Si ces fiches sont désignées par leur numéro local, ces références sont changées en identifiants globaux.

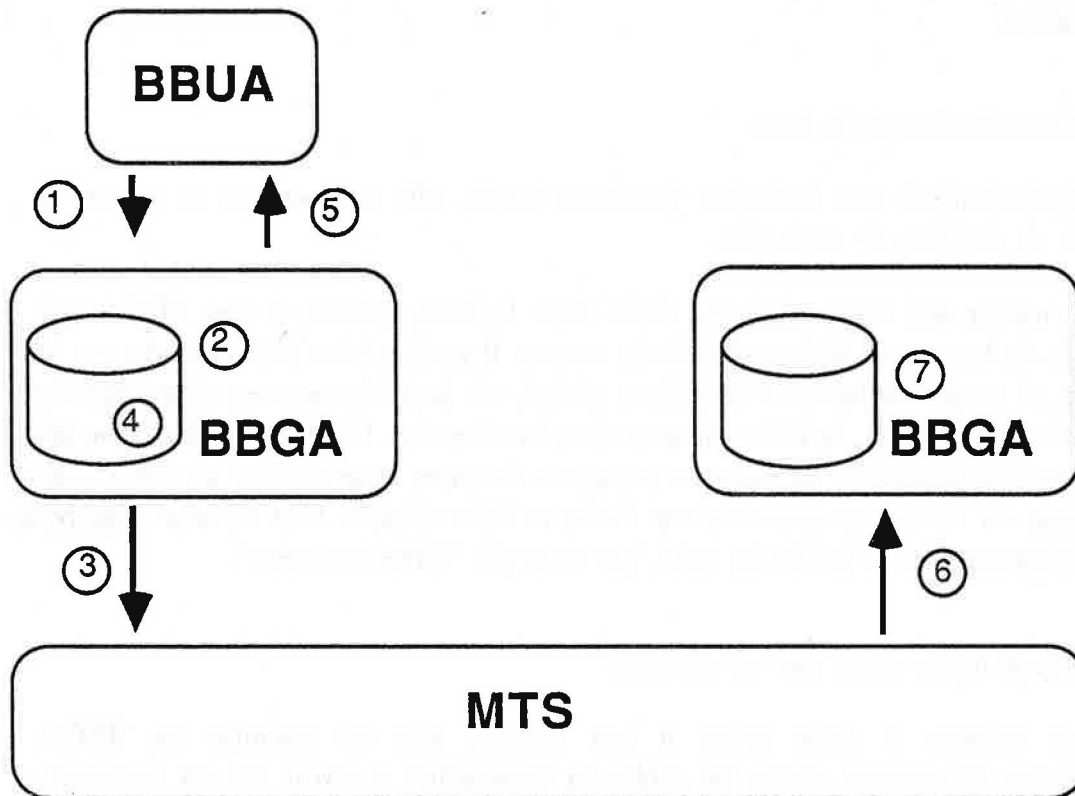
Si le contrôle est satisfaisant, le BBGA complète le descripteur de la fiche en ajoutant la date de prise en compte, l'identifiant et l'état : normal.

Le BBGA génère un message "diffusion de fiche" qui contient la fiche créée, complétée et débarrassée de toute référence locale. Ensuite, il émet une *Demande de dépôt* vers le MTA. Les destinataires du message sont tous les BBGA coopérants de la conférence.

Le BBGA donne la date de péremption de la fiche. Si une proposition a été faite par l'auteur de la fiche, elle est limitée à la durée maximale mémorisée par le BBGA. Sinon, le BBGA choisit la date par défaut qu'il connaît.

La fiche est insérée dans la base et un compte rendu est généré pour le BBUA. Il sera transmis à l'auteur.

Sur chaque BBGA destinataire, la fiche est prise en compte, les relations sont répercutées. Si une fiche liée est inconnue, une fiche fictive est créée.



1 : la fiche de l'auteur est soumise au BBGA

2 : le BBGA contrôle les droits de l'auteur, les liens de la fiche  
( en cas de problèmes, --> 5 ),  
il complète le descripteur et remplace toutes les références locales

3 : le BBGA diffuse la fiche vers l'ensemble des BBGA de la conférence

4 : le BBGA ajoute la date de péremption et insère la fiche dans sa  
version de la base.

5 : un compte rendu est émis vers le BBUA,

6 : chaque BBGA reçoit la fiche,

7 : et l'insère dans sa base locale.

Fig III. 18 : Insertion d'une fiche par un écrivain

### 3.7.3 Ajout d'une fiche par un usager non écrivain

Lorsque l'auteur de la fiche n'est pas écrivain, la fiche doit être soumise à un éditeur. Le traitement effectué par le BBGA est le suivant (Fig. III.19).

- Contrôle des droits de l'utilisateur qui doit être non écrivain.
- Vérification de l'existence des fiches liées, et remplacement de toutes les références locales.
- En cas de problèmes pendant les vérifications, un compte-rendu négatif vers le BBUA de l'auteur termine le traitement du BBGA.
- Autrement, le BBGA teste si l'auteur a indiqué un éditeur. Dans la négative, le BBGA choisit l'éditeur par défaut. Cet éditeur est donné par l'administrateur général ou éventuellement par l'administrateur local du BBGA.
- Si l'éditeur désigné est un abonné du BBGA, ce dernier vérifie ses droits. Si l'utilisateur est effectivement éditeur, son BBUA reçoit la demande d'imprimatur. Sinon, un compte rendu négatif est transmis au BBUA de l'auteur.
- Si l'éditeur appartient à un BBGA distant, la demande est envoyée à ce BBGA grâce au système de transfert de message.

Lorsque le BBGA de l'éditeur distant reçoit le message, il vérifie successivement :

- L'existence du BBGA expéditeur dans la liste des sites de la conférence, en cas de problème une indication d'anomalie est émise vers le BBUA de l'administrateur local et le traitement est stoppé.
- L'existence et les droits de l'utilisateur désigné comme éditeur, si l'utilisateur est inconnu ou incapable, une indication d'anomalie est envoyée à l'administrateur local et un message "Réponse-éditeur" est retourné au BBGA expéditeur pour lui signaler l'anomalie.
- Le BBGA transmet la demande d'imprimatur à l'éditeur concerné.

Lorsqu'un éditeur reçoit via son BBUA une demande d'imprimatur, il l'accepte ou la refuse, dans tous les cas il peut donner un commentaire. L'avis de l'éditeur est transmis à son BBGA.

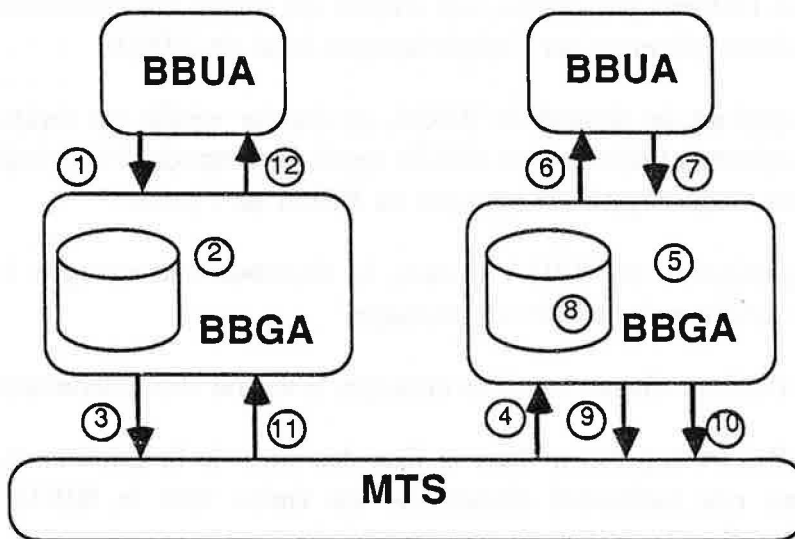
Si la réponse est favorable :

- le BBGA ajoute le nom d'éditeur au descripteur de la fiche et l'identifiant de la fiche, fixe l'état de la fiche à normal mais ne vérifie pas l'existence des fiches liées. Ceci a été fait sur le site de l'auteur. En raison des incohérences

entre versions, certaines des fiches liées risquent d'être inconnues sur le site de l'éditeur alors qu'elles existent dans la version du BBGA de l'auteur.

- Le BBGA de l'éditeur diffuse un message "Diffusion de fiche" vers tous les BBGA de la conférence qui ajouteront la fiche dans leur base.
- Il fixe la date de péremption selon la procédure décrite pour l'ajout de fiches. La fiche est alors insérée dans la base locale.

Quelle que soit la réponse de l'éditeur, un message "Réponse d'éditeur" est envoyé vers le BBGA de l'auteur.



- 1 : la fiche de l'auteur est soumise au BBGA
- 2 : le BBGA contrôle les droits, les fiches liées et l'éditeur
- 3 : si l'éditeur n'est pas local, la demande est émise vers son BBGA
- 4 : le BBGA de l'éditeur reçoit le message
- 5 : le BBGA contrôle l'existence et les droits de l'éditeur
- 6 : le BBGA transmet la demande au BBUA de l'éditeur
- 7 : la réponse de l'éditeur parvient à son BBGA
- 8-9 : si la réponse est positive, le BBGA complète, diffuse et insère la fiche
- 10 : la réponse de l'éditeur est envoyée au BBGA de l'auteur
- 11-12 : le BBGA reçoit la réponse et la transmet au BBUA

Fig III.19 : Insertion d'une fiche par un auteur

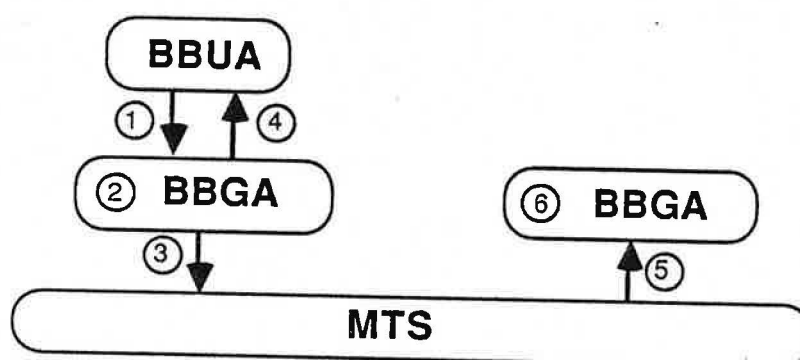
### 3.7.4 Effacement d'une fiche

Effacer une fiche ne la fait pas disparaître de la base, seul son contenu est supprimé. La fiche demeure ce qui sauvegarde la structure de la base. Les fiches liées par des relations qui supportent l'héritage de l'effacement sont elles aussi effacées. Une fiche est effaçable par son auteur, l'administrateur local ou l'administrateur général.

L'utilisateur transmet sa demande au BBGA via son BBUA (Fig. III.20). Il désigne la fiche concernée par un numéro local ou un identifiant général. Le BBGA vérifie l'existence de la fiche et les droits de l'utilisateur. En cas de problème, un compte rendu négatif accompagné de la cause du rejet est retourné au BBUA.

Si les contrôles le permettent, la fiche est effacée localement ainsi que sa descendance selon les types de filiation. Si ce n'est pas fait par le demandeur, le BBGA donne l'identifiant global de la fiche et il diffuse la requête vers les autres BBGA. Un compte rendu positif est envoyé à l'initiateur de la demande.

Lorsqu'il reçoit la requête d'effacement, chaque BBGA distant vérifie l'appartenance du BBGA émetteur à la conférence. Si la fiche concernée est inconnue, une fiche fictive est créée, son état est *fictive à effacer*. Sinon, la fiche réelle est effacée. Le BBGA propage l'effacement dans la descendance de la fiche traitée. Lorsqu'une fiche fictive est rencontrée au cours de cette exploration, son état devient : *fictive à effacer*.



- 1 : la demande d'effacement est transmise au BBGA
- 2 : le BBGA vérifie les droits du demandeur et l'existence de la fiche concernée.  
Si tout est conforme, la fiche est effacée ainsi que les descendantes impliquées.
- 3 : si tout se passe bien, la demande d'effacement est diffusée.
- 4 : le BBGA du demandeur reçoit un compte-rendu.
- 5 : la demande est reçue par les autres BBGA de la conférence
- 6 : chaque BBGA traite la requête, effacement et héritage.

Fig III.20 Effacement d'une fiche



The first part of the report is devoted to a description of the experimental conditions and the results obtained. The second part is devoted to a discussion of the results and a comparison with the theoretical predictions. The third part is devoted to a conclusion and some suggestions for further work.

The first part of the report is devoted to a description of the experimental conditions and the results obtained. The second part is devoted to a discussion of the results and a comparison with the theoretical predictions. The third part is devoted to a conclusion and some suggestions for further work.

The first part of the report is devoted to a description of the experimental conditions and the results obtained. The second part is devoted to a discussion of the results and a comparison with the theoretical predictions. The third part is devoted to a conclusion and some suggestions for further work.

The first part of the report is devoted to a description of the experimental conditions and the results obtained. The second part is devoted to a discussion of the results and a comparison with the theoretical predictions. The third part is devoted to a conclusion and some suggestions for further work.



---

<b>LES DIFFICULTES DE L'INTERFACE</b>	<b>83</b>
Etude récente	83
Complexité des logiciels	84
Grande diffusion des produits	84
Indépendance vis-à-vis du système	85
Accès au produit	86
Personnalisation de l'interface	86
Monde de l'utilisateur et monde informatique	87
Absence d'outils pour les concepteurs	87
<b>LES NOUVELLES INTERFACES UTILISATEUR</b>	<b>88</b>
WYSIWYG	88
Les objets	89
Les menus	90
La métaphore du bureau	91
Le matériel et logiciels nécessaires	91
Spécification de l'interface	92
Boîtes à outils	93
Unité logique	93
Terminal virtuel	94
Entrées-Sortie de haut niveau	96
Les intégrateurs	97
<b>LE BUREAU CRMM</b>	<b>98</b>
Services à offrir	98
Station de travail	99
UA éclaté	100
Les objets internes de l'interface CRMM	101
La liste des numéros locaux des fiches non lues,	101
La liste des fiches précédentes	102
La liste des fiches sélectionnées	102
La liste des fiches marquées	103
La liste des alias des sites	103
Le contenu de la boîte	103
La description de la conférence	105
La description de l'ensemble des conférences	105
Les fiches disponibles localement	105
Les objets d'affichage	106
La fenêtre de lecture	106
La fenêtre de sélection de fiches	107
La fenêtre de résultat	108
La fenêtre des marques	109
La fenêtre de la boîte aux lettres	109
La fenêtre de copie	110
La fenêtre d'état	110
La fenêtre de création de fiche	111



## Chapitre 4

### INTERFACE UTILISATEUR

De plus en plus de personnes emploient des outils informatiques. Ceux-ci sont extrêmement variés, puissants et complexes. Mais ce développement est freiné par des difficultés de mise en œuvre et d'utilisation. Elles ne sont pas seulement dues à des problèmes d'algorithmes mais dépendent également de l'interface utilisateur.

**L'interface utilisateur ou interface interactive est le logiciel chargé de la communication d'informations entre un système informatique et l'utilisateur.**

Grâce à elle, le système doit être simple d'apprentissage et d'emploi, efficace et agréable d'utilisation.

Les premiers paragraphes de ce chapitre présentent les difficultés de ce domaine, puis les solutions envisageables sont expliquées et enfin les propositions retenues dans le projet CRMM sont détaillées.

#### 1 LES DIFFICULTES DE L'INTERFACE

##### 1.1 ETUDE RECENTE

Les problèmes de communication entre l'homme et la machine ont été abordés récemment. Ce domaine est complexe, il déborde le cadre de l'informatique et s'étend à des disciplines telles que l'ergonomie ou la psychologie. Cette pluridisciplinarité pose des problèmes de transfert de connaissances entre spécialistes. Aussi, les débuts sont hésitants et il ne faut pas s'étonner des contradictions dans les intentions.

Les travaux sont donc récents. Sans doute, a-t-il fallu attendre que l'algorithmique des programmes atteigne une certaine qualité pour que les faiblesses de l'interactivité apparaissent. Les programmeurs avaient suffisamment de problèmes par ailleurs pour s'occuper de ce qui paraissait n'être que l'emballage final sans influence sur les qualités intrinsèques des programmes. En outre, les utilisateurs étaient peu nombreux et devaient être informaticiens, ce qui permettait de reporter les problèmes de mise en œuvre en aval et de les laisser à la charge de l'utilisateur final.

Toutefois, il est certain que les progrès dans le dialogue sont étroitement liés à la plus large diffusion des outils informatiques. Ils l'ont rendue possible et ont bénéficié d'une forte demande de la part des nouveaux utilisateurs déconcertés par leurs logiciels d'application.

## 1.2 COMPLEXITE DES LOGICIELS

Les logiciels proposés aux grands publics deviennent de plus en plus complexes. Désormais, ils abordent des domaines qui semblaient réservés aux spécialistes. Par exemple, les simples gestionnaires de fichiers laissent la place à de véritables systèmes de gestion de bases de données relationnelles. Quelle que soit la qualité du produit, leur utilisation n'est pas facile par nature même. A ce propos, il faut signaler qu'une simplification excessive de l'interface risque d'altérer le produit.

## 1.3 GRANDE DIFFUSION DES PRODUITS

Les logiciels doivent être utilisables par le plus grand nombre, les connaissances nécessaires à leur emploi doivent être minimales. Parallèlement, une large diffusion rend difficile la formation et l'assistance aux utilisateurs, d'autant plus que les contraintes économiques tendent à limiter les coûts, ce qui signifie minimiser le temps de formation et le suivi des clients.

Le public est très varié, d'origine, de culture, de profession et de niveau informatique différents. Il devient de plus en plus difficile pour les concepteurs d'applications de prévoir les demandes et préférences des utilisateurs. Ceci est aggravé par la large diffusion des produits qui éloigne encore plus les réalisateurs des utilisateurs finals.

Les produits proposés sont variés. Ils couvrent beaucoup de domaines. Pendant une "période" de travail, l'utilisateur met en œuvre différents outils. Par exemple, il se sert d'un traitement de texte, d'un tableur, d'un éditeur graphique ou d'un logiciel plus spécifique. Cette diversité pose les problèmes suivants :

Tout d'abord, il faut connaître plusieurs produits. Le temps d'apprentissage augmente d'autant.

Les différences d'emploi entre les produits sont source de confusions. Par exemple, celles-ci sont fréquentes lorsqu'une même touche du clavier n'a pas des effets semblables d'un produit à l'autre.

Les logiciels risquent d'être incompatibles quant au format des données traitées. Or il est fréquent qu'ils aient à échanger des informations. Par exemple, composer un document nécessite la coopération d'un traitement de texte, d'un tableur et d'un éditeur graphique.

Les produits doivent partager les ressources de la machine. La gestion de l'espace disque est généralement bien traitée par les systèmes d'exploitation actuels, en revanche le partage de la mémoire principale pose plus de problèmes, quant à l'écran, son utilisation simultanée par plusieurs applications est loin d'être courante. Cette concurrence d'accès aux ressources est d'autant plus difficile à gérer que les produits ne sont pas censés se connaître. Ils n'ont pas obligatoirement la même origine. Le format des données évoqué précédemment, risque fort de ne pas être la seule différence de conception.

Enfin, l'étude séparée de l'interface de chaque produit pris isolément ne répond pas aux problèmes d'ergonomie posés par l'activité globale de l'utilisateur, comme le séquençage des tâches, les interruptions inévitables. Par exemple, que propose le produit lorsque l'activité est interrompue par une autre plus urgente ? Le système devrait être capable de sauvegarder, puis de restaurer l'état de la tâche courante.

#### 1.4 INDEPENDANCE VIS-A-VIS DU SYSTEME

La large diffusion d'un logiciel exige qu'il soit indépendant de la machine sur laquelle il est exploité. Ce problème ne concerne pas seulement l'interface, mais la totalité du produit. D'ailleurs, les traitements tels que les calculs, manipulations de chaînes, tests, entrées-sorties sur fichiers ou terminaux classiques, sont rendus universels par des langages de haut niveau comme C. En revanche, l'interactivité plus élaborée, risque d'être fortement liée aux périphériques utilisés. Il suffit de voir les problèmes posés par les cartes graphiques des compatibles PC. Cette difficulté est résolue par le concepteur qui doit prévoir des échanges avec le maximum de types de périphériques, ou alors, elle est laissée à la charge de l'utilisateur, qui doit posséder les périphériques requis par ses applications ou limiter son choix d'outils à ceux compatibles avec son matériel.

Les problèmes évoqués ci-dessus sont amplifiés par les modifications du système informatique dans lequel évoluent les applications. Les raisons du changement sont multiples.

- Les utilisateurs éprouvent le besoin de nouveaux outils.
- Le matériel est périmé.
- La puissance des outils devient insuffisante.
- Des outils plus performants sont disponibles.

Ces changements de machines, de systèmes d'exploitation ou de logiciels d'application sont fréquents. Ils rendent d'autant plus coûteux la formation et l'apprentissage aux produits. L'adaptation des personnes à ces outils doit être rapide. Ce problème s'apparente à celui de l'individu qui exploite simultanément différents outils. Là aussi, les différences entre versions doivent être minimales pour que le passage de l'une à l'autre soit le moins douloureux possible.

## 1.5 ACCES AU PRODUIT

Il est vraisemblable que plusieurs personnes peuvent être amenées à partager le même logiciel. Pour des raisons financières (coûts d'acquisition, de maintenance, de stockage), il n'est guère envisageable de posséder autant de copies du produit que d'utilisateurs potentiels.

Parallèlement à la volonté de limiter le nombre d'exemplaires d'un logiciel, il faut pourtant veiller à ce que les utilisateurs aient un accès aisé aux produits qui les intéressent. Dans une organisation suffisamment importante, les outils sont disponibles sur différentes machines et les usagers accepteraient mal d'incessants déplacements de l'une à l'autre. De plus, la coopération devient encore plus délicate entre outils de différentes machines.

## 1.6 PERSONNALISATION DE L'INTERFACE

Les qualités de l'interface dépendent des individus auxquels elle s'adresse. Un dialogue fortement assisté, avec multiples messages d'explication et demandes de confirmation est apprécié par un novice, mais devient pénible pour l'utilisateur confirmé. De même, lors de la phase d'apprentissage, l'utilisateur accepte d'avoir à consulter sans cesse la documentation. En revanche, un individu qui se sert fréquemment d'un outil refuse de recourir sans arrêt au mode d'emploi et est gêné par une aide trop envahissante (demandes de confirmation, messages d'aide).

Les remarques précédentes mettent en évidence le point suivant. Un usager habituel n'est pas nécessairement expert dans son utilisation. En effet, si un expert connaît bien toutes les fonctionnalités de l'application, en revanche, l'utilisateur peut n'être intéressé que par un sous-ensemble des fonctions de l'application.

Prenons comme exemple un formateur de texte. Un utilisateur qui se limite à des textes simples n'utilise qu'un jeu de commandes réduit. S'il accepte bien d'avoir à consulter la notice pour des options complexes ou peu courantes telles que la génération d'un index ou d'un sommaire, il refuse cette démarche, pour des commandes usuelles comme le saut de lignes ou la création de paragraphes.

Compte tenu du grand nombre d'outils qu'une personne est susceptible d'utiliser et des fréquentes évolutions des outils, la classe des experts semble réduite hormis pour les applications très spécifiques. La plupart des outils d'intérêt général comme le sont les traitements de texte et la messagerie sont destinés à des utilisations fréquentes pour lesquelles les opérations classiques sont à privilégier.

## 1.7 MONDE DE L'USAGER ET MONDE INFORMATIQUE

Le grand nombre d'utilisateurs potentiels laisse prévoir qu'une minorité d'entre eux ait des connaissances en informatique. De plus, une formation préalable à l'informatique semble exclue. Donc, les concepteurs ne doivent pas compter sur des éventuelles compétences informatiques de la part des utilisateurs de leurs produits. Cela signifie que le produit doit minimiser les références aux objets informatiques, que ce soit dans sa description, dans son mode d'emploi et dans le dialogue avec l'utilisateur. Pour la même raison, la structure apparente de l'application ne doit pas dépendre de contraintes informatiques. Certes, celles-ci sont inévitables et influent sur l'application, mais l'interface doit les cacher le plus possible et limiter leurs effets sur l'emploi du produit.

## 1.8 ABSENCE D'OUTILS POUR LES CONCEPTEURS

Alors que les exigences augmentent, les concepteurs d'applications interactives sont désarmés. Aucun outil traditionnel n'est adapté à la réalisation de ces programmes. Ces carences apparaissent dans la gestion des entrées-sorties dont le niveau d'abstraction est trop faible. Par exemple, C ne permet que la manipulation de simples chaînes de caractères.

Les problèmes décrits précédemment, sont dépendants les uns des autres. Aussi, les propositions faites par la suite, ne répondent pas individuellement à chacun d'eux. En outre, des solutions triviales "de bon sens" ne sont pas détaillées ici. Par exemple, le fait que l'utilisateur n'ait que deux mains et qu'il ressente une gêne pour frapper plus de trois touches simultanément, est supposé acquis.



La thèse n'avait pas pour but de faire évoluer les connaissances en ce domaine, mais plutôt de faire le point sur les outils et options actuels, et d'en faire bénéficier une application.

Parmi les progrès faits dans l'interface, il faut distinguer ceux qui portent sur les outils matériels et logiciels propices aux développements de bonnes interfaces, ceux qui s'attachent aux méthodes de développement et enfin ceux qui font évoluer le concept même d'interface utilisateur, en dehors des problèmes de réalisation.

## 2 LES NOUVELLES INTERFACES UTILISATEUR

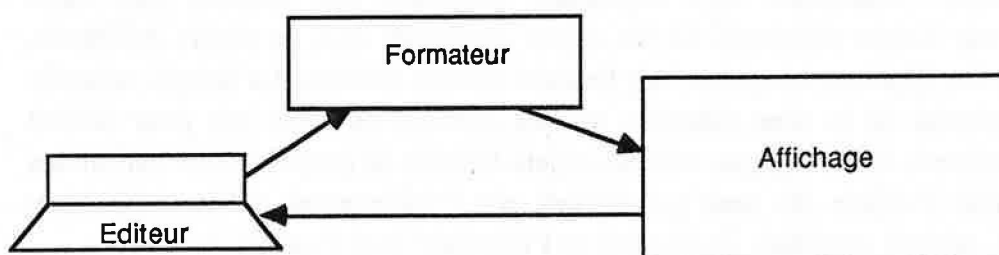
Les plus gros progrès ont pour origine les travaux de l'équipe d'A. Kay de Xerox Parc qui dès 1971 lança une étude sur un ordinateur personnel utilisable par tout le monde. Une des caractéristiques de ce projet fut d'être basé sur les possibilités des ordinateurs à venir, et non pas sur celles des ordinateurs de l'époque. Leurs travaux ont principalement débouché sur le langage et l'environnement de programmation Smalltalk, et sur la station de travail Star.

### 2.1 WYSIWYG

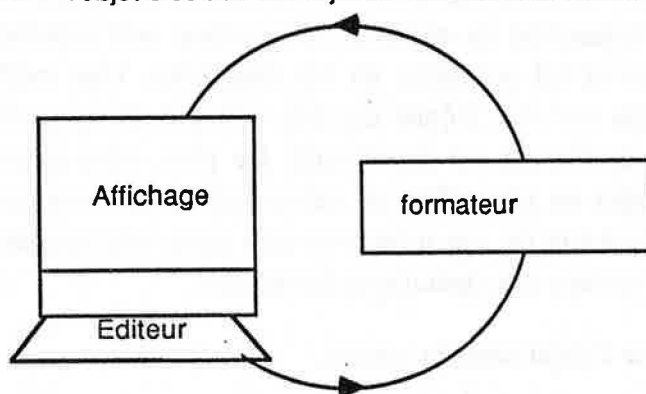
Une bonne interactivité entre le système et l'utilisateur demande que ce dernier ait une vision correcte de l'objet manipulé. Une solution est de faire apparaître à l'écran une représentation la plus exacte possible. Par exemple, un traitement de texte affiche le texte tel qu'il sera effectivement imprimé, en tenant compte des polices de caractères choisies, notamment si elles sont proportionnelles, de la justification désirée et des espacements entre mots qu'elle implique. Cette caractéristique des applications interactives est connue sous le label : WYSIWYG pour *What You See Is What You Get*.

Cette volonté se heurte aux réalités de puissance de traitement et d'affichage des machines actuelles. Une modification dans la justification d'un texte est une opération coûteuse. Il ne faut pas tomber dans l'excès *What You Get Is Only What You See*. En effet, les limites du matériel d'affichage ne doivent pas provoquer l'abandon de toutes les opérations dont les effets ne sont pas visualisables. Par exemple, les possibilités des imprimantes à laser, avec leur précision au trois centièmes de pouce doivent être prises en compte même si la précision des écrans graphiques est plus faible.

Le débat actuel entre WYSIWYG et traitement par lots (*batch*) (Fig. IV.1), c'est-à-dire entre un système fortement interactif, plus agréable d'emploi et un système aux fonctionnalités plus puissantes, semble tourner court. En effet, les progrès du matériel laissent envisager un avenir où l'interactivité n'est plus limitée par les possibilités du système [BOHN87].



l'objet d'édition et l'objet de résultat sont différents



Lorsque le traitement est interactif, le résultat sert d'objet d'édition

Fig IV.1 : Traitement WYSIWYG et traitement par lots

## 2.2 LES OBJETS

Une notion importante répandue par A. Kay a été évoquée dans les paragraphes précédents. Il s'agit des objets. Cette notion est aujourd'hui de plus en plus largement répandue. Elle permet une description différente des traitements à réaliser. Auparavant, un programme était vu comme une séquence d'actions à effectuer sur des opérandes. Par exemple, l'instruction en assembleur : *LDA 5260* signifie charger l'accumulateur (ACTION) avec le contenu de l'adresse 5250 (OPERANDE). Cette démarche où le verbe précède l'objet n'est pas naturelle pour l'utilisateur quelconque. Pour lui, la démarche sélection d'un objet, puis opération sur cet objet est plus habituelle.

La plupart des langages de commande en vigueur, pour les systèmes d'exploitation par exemple, fonctionne selon le premier schéma. Ainsi, la commande ERASE A:TOTO.EXE détruit le fichier a:TOTO.EXE. En revanche, les utilisateurs sélectionnent les fichiers concernés, puis demandent leur destruction.

En outre, l'expression 5250 représente seulement une adresse, sans autre information sur l'objet référencé. Or les objets manipulés sont de nature différente, par exemple les applications gèrent des fenêtres ou des fichiers. La simple connaissance de l'adresse de la zone mémoire qui les contient ne suffit pas pour décrire leur comportement. Les langages orientés objets laissent le programmeur définir ses propres classes d'objets. Ils sont caractérisés par l'information qu'ils contiennent (ou champs), actions possibles (méthodes) et l'interface avec l'extérieur.

Cette notion d'objets est intéressante pour la conception d'applications mais elle doit aussi influencer sur l'apparence du système. L'interface doit représenter les objets connus par l'utilisateur et lui permettre de les manipuler. Une méthode de représentation graphique basée sur des icônes est très efficace et compréhensible pour l'utilisateur. Le dessin de l'icône est significatif. De plus, elles peuvent être nommées. Elle évitent à l'utilisateur un gros effort de mémorisation de commandes. En effet, la manipulation de l'objet (ou de son icône) est elle aussi très imagée. Sur le Macintosh, détruire un objet revient aux opérations suivantes :

- sélection en cliquant sur l'objet avec la souris,
- déplacer l'objet jusqu'à la corbeille.

### 2.3 LES MENUS

Les actions que supportent les objets sont accessibles grâce à des menus. Certes, ils sont moins souples et moins puissants que les langages de commandes. Par exemple, ils semblent inadaptés à la direction d'un système d'exploitation comme UNIX, avec énormément de commandes possibles, la plupart demandant des paramètres. En revanche, pour un logiciel d'application au jeu de commandes plus réduit, l'approche par menus est intéressante car elle évite à l'utilisateur les problèmes de syntaxe des langages de commandes et elle permet une visualisation des commandes disponibles, ce qui supprime des efforts de mémorisation [HODGSON85]. Un problème apparaît lorsque le nombre des commandes devient important. Les menus amènent alors une hiérarchisation qui risque d'être lourde et contraignante. Ceci est d'autant plus pénible si la structure imposée par les menus ne correspond pas à celle concevable par l'utilisateur. L'appel d'une commande donnée nécessite alors un parcours de l'arborescence. Des menus dits menus déroulants (*popup menu*) lèvent en partie ces difficultés. En effet, ils permettent un accès rapide à toutes les commandes de l'arborescence qu'ils décrivent. Tant que les études sur la langue naturelle et la reconnaissance vocale n'auront pas porté leurs

fruits [HAYES85], l'approche par menus semble préférable pour la plupart des logiciels d'application.

## 2.4 LA METAPHORE DU BUREAU

Ce que nous venons de décrire, permet donc de manipuler facilement des objets informatiques. Une difficulté demeure toutefois tant que l'utilisateur ne comprend pas la nature et le rôle des objets qui lui sont présentés. Pour remédier à cela, il faut éviter les allusions au monde informatique, et au contraire, faire une analogie entre les objets informatiques et les objets d'un monde familier à l'utilisateur, par exemple, le monde du bureau.

Ainsi, on parle plus de fichiers mais de documents, plus de répertoires mais de dossiers. Cette analogie dépasse le cadre du vocabulaire. Elle doit être sensible dans la présentation des objets. Sur le Macintosh, l'écran est une table de bureau, avec l'icône de la poubelle et les icônes des rangements. L'utilisateur ouvre ces derniers où se trouvent les dossiers et les documents. Il affiche les documents qui se superposent tels des feuilles de papier posées sur un bureau. Cet apprentissage de techniques et de notions nouvelles à l'aide d'analogies avec un monde connu et familier est un procédé courant [HOUSTON85].

## 2.5 LE MATERIEL ET LOGICIELS NECESSAIRES

Ce qui précède demande au moins :

- un écran graphique de bonne résolution,
- un outil de pointage comme une souris, un écran tactile ou un photostyle,
- un système de fenêtrage de l'écran,

Il est évident que pour gérer ce système, une puissance de traitement et une capacité mémoire suffisantes sont indispensables. Heureusement, le matériel a progressé pour offrir des composants à des prix suffisamment bas pour que la large diffusion de ces produits soit envisageable, ce qui réduit d'autant leur coût.

## 2.6 SPECIFICATION DE L'INTERFACE

Les fonctionnalités d'un système sont disponibles grâce aux primitives qu'il offre. Chacune comporte un certain nombre de paramètres. L'interface doit permettre à l'utilisateur de faire appel aux services, de fournir les informations nécessaires et de prendre connaissance des résultats. Cette première définition est insuffisante. Se limiter à celle-ci conduirait à des produits où l'utilisateur est quantité négligeable. En effet, l'aspect de l'interface risque d'être trop influencé par les primitives offertes par le système. Or, le choix des primitives dépend beaucoup de contraintes informatiques dont l'utilisateur n'a que faire (simplification des programmes, diminution du nombre de cas possibles).

Il faut au contraire que l'interface soit orientée vers les besoins de l'utilisateur. Elle doit prendre en compte sa tournure d'esprit et la perception qu'il peut avoir du produit. L'interface ajoute un service au système, elle gère un dialogue adapté à l'utilisateur et fait appel aux primitives. Ce service doit être spécifié indépendamment des fonctionnalités du système. Ce découpage est nécessaire, mais il est aussi très profitable car les concepteurs des services de base n'ont plus à se soucier de l'utilisateur, ce qui facilite grandement leur tâche. Il permet également de spécifier des interfaces différentes pour un même système fonctionnel. Il n'existe donc pas *a priori* de relations directes entre les primitives du dialogue avec l'utilisateur et celles d'accès aux services (Fig. IV.2).

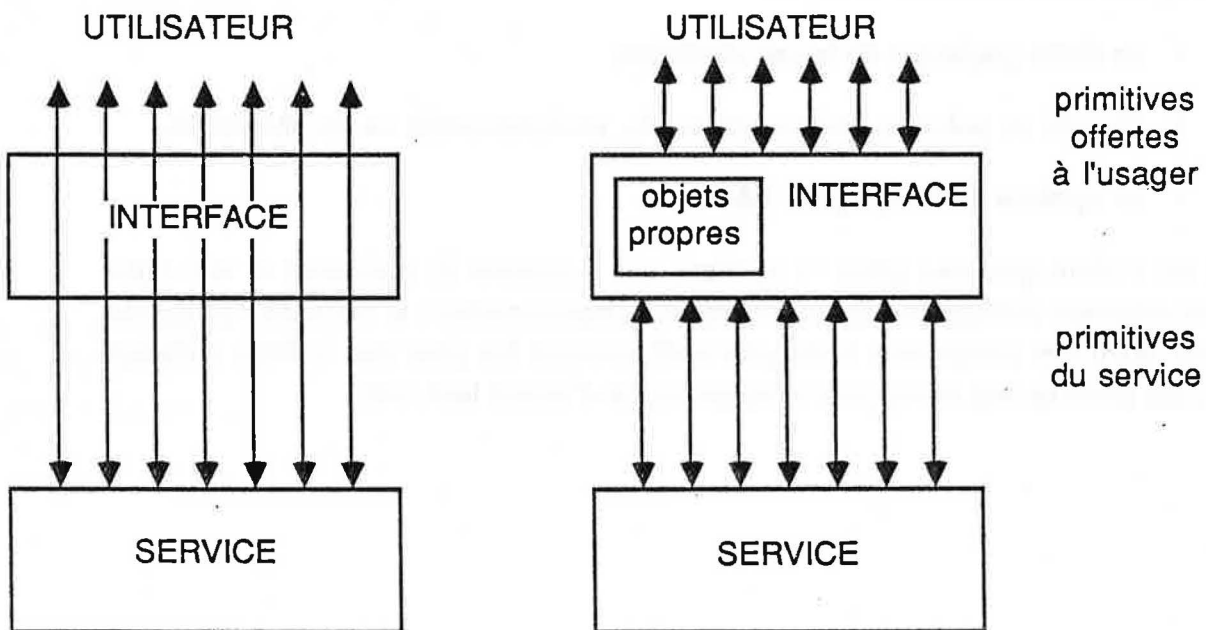


Fig IV.2 : Découpage service et interface

## 2.7 BOITES A OUTILS

Maintenant que les buts de l'interface sont définis, que les spécifications sont établies, il s'agit d'offrir des outils aux réalisateurs. A eux aussi, il faut masquer la complexité des programmes nécessaires à l'interactivité. Pour cela, ils disposent d'objets dont le niveau d'abstraction est compatible avec celui des objets à offrir à l'utilisateur. Les outils nécessaires à la création et manipulation de ces objets font partie d'une boîte à outils.

## 2.8 UNITE LOGIQUE

Le besoin d'une indépendance de l'application vis-à-vis du matériel a été signalé plus haut. Pour cela, les applications ne dialoguent plus avec un terminal réel, mais avec des unités logiques [COUTAZ86]. Ces dernières se chargent de la gestion des unités physiques. Des propositions de normes comme GKS<sup>1</sup>, répartissent ces unités logiques en classes. Pour chacune d'elles, des attributs et des primitives sont définis.

Par exemple, une souris, unité physique, est vue par l'application comme deux unités logiques :

- un "localisateur" qui représente le suiveur et fournit deux coordonnées,
- un "choix" pour les touches qui donne un entier.

Une unité logique de sortie comprend la primitive "Rectangle(1, 1, 10, 10)" qui correspond au tracé d'un rectangle. Elle se charge de traduire cette requête et de la transmettre au périphérique concerné.

Si le matériel change, les programmes d'applications n'auront pas être réécrits. Il suffit que la boîte à outils offrent les pilotes (*drivers*) des nouveaux matériels capables d'interpréter les primitives logiques.

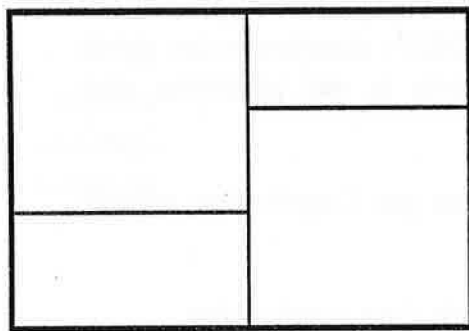
---

1 GKS : Graphic Kernel System est un standard pour l'interface de périphériques graphiques

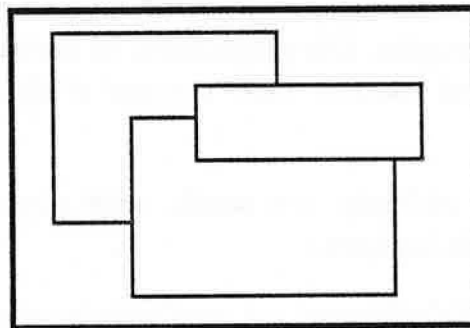
## 2.9 TERMINAL VIRTUEL

Les applications doivent partager les ressources. Comme les applications ne sont pas censées se connaître, il est difficile aux concepteurs de gérer cette concurrence. Aussi, la boîte doit offrir aux applications des terminaux virtuels qui lui donne l'impression d'être l'unique propriétaire du terminal réel. Pour cela, ils sont constitués d'un ensemble d'unités logiques. L'une d'elle est fréquemment employée, il s'agit de la fenêtre.

Souvent rectangulaire, elle sert d'écran réel à l'application. Les fenêtres présentent sur l'écran physique se recouvrent comme dans Smalltalk ou créent une partition de l'écran, option retenue par l'environnement Cedar de Xerox (Fig. IV.3).



Partition de l'écran



Recouvrement de fenêtres

Fig IV.3 : Partitionnement et recouvrement

Le programmeur n'a pas à se soucier du partage de l'écran physique. Les recouvrements et changements de tailles de fenêtres sont à la charge du système qui signale à l'application si la fenêtre doit être régénérée (Fig. IV.4).

Si les cadres sont toujours disponibles pour l'affichage, les autres unités logiques qui composent le terminal virtuel ne le sont que lorsque l'application est prioritaire. Ainsi, chaque application dialogue avec une unité logique qui représente le clavier. Or, lorsque l'utilisateur procède à des entrées via ce matériel, une seule unité logique est concernée. Pour la désigner, l'utilisateur peut par exemple promener le curseur attaché à la souris dans la fenêtre de l'application choisie (Fig. IV.5).

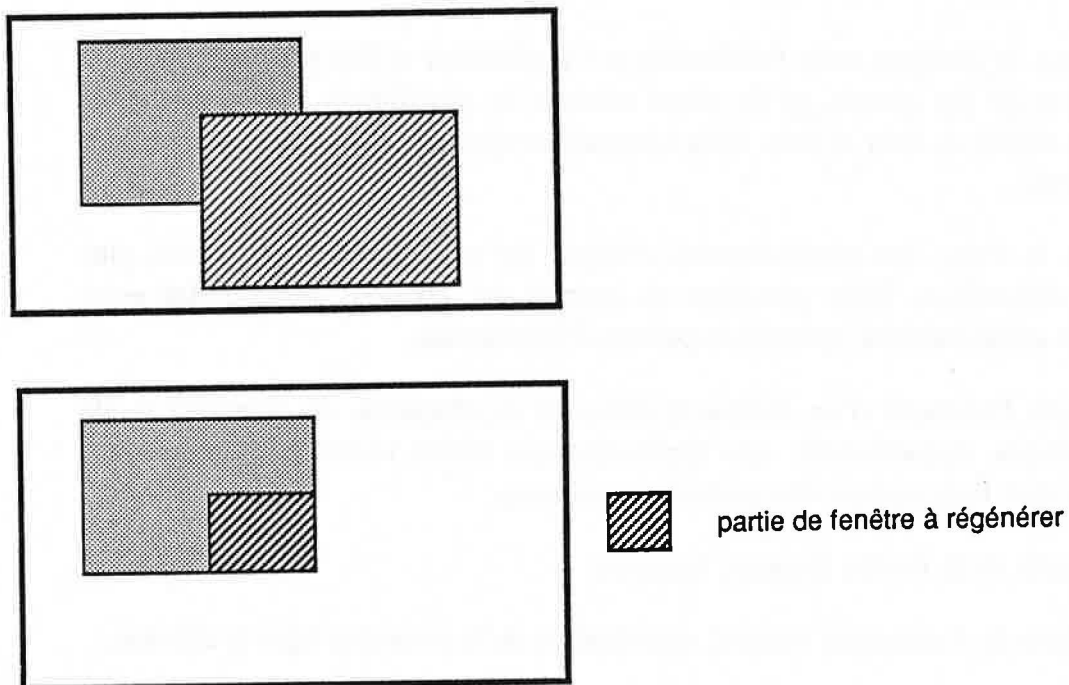


Fig IV.4 : Régénération de fenêtre

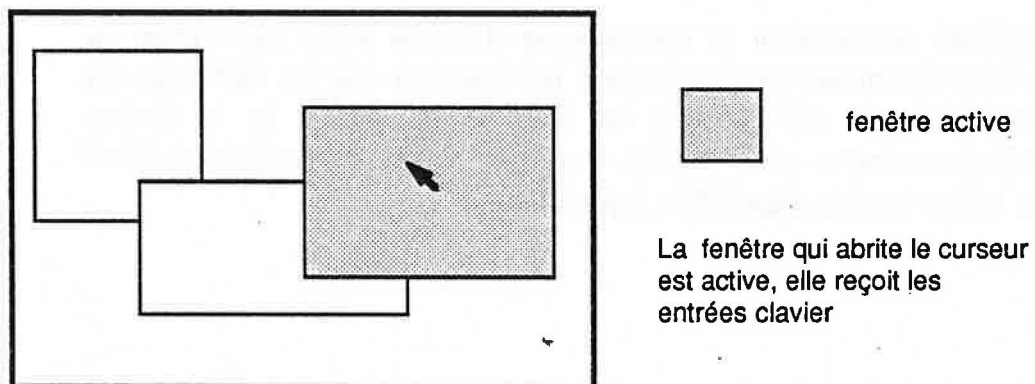


Fig IV.5 : Mise en évidence de la fenêtre active



## 2.10 ENTREES-SORTIE DE HAUT NIVEAU

Actuellement, le dialogue entre l'utilisateur et l'application se fait grâce à des objets informatiques de bas niveau, qu'ils soient textuels ou graphiques. Certes, ils représentent des objets de haut niveau, mais intrinséquement, ils demeurent extrêmement rudimentaires.

Dans le futur, les entrées/sorties d'objets de plus haut-niveau seront plus largement disponibles. Elles prendront en compte des relations structurelles entre objets et un comportement syntaxique pendant l'interaction.

Prenons l'exemple d'un document composé de chapitres, de paragraphes, de lignes de textes. Actuellement, une application qui désire visualiser le texte dans une fenêtre doit tenir compte des paramètres suivants :

- La taille de la fenêtre (largeur, hauteur).
- Position de l'ascenseur vertical, significative de la première ligne à afficher.
- Position de l'ascenseur horizontal, significative du début d'affichage de chaque ligne.
- Choix de la police de caractères.
- Dès qu'un de ces paramètres change, l'application doit recalculer la partie de texte appartenant à la fenêtre d'affichage et écrire chaque chaîne de caractères visible (Fig. IV.6).

Si les entrées-sorties permettaient la manipulation d'entités telles que l'objet de classe "texte", le programmeur de l'application ne donnerait que les références du texte à visualiser (adresse en mémoire ou nom de fichier) et de la fenêtre concernée. Malheureusement, cette facilité n'est pas encore suffisamment développée dans les boîtes à outils disponibles aujourd'hui.

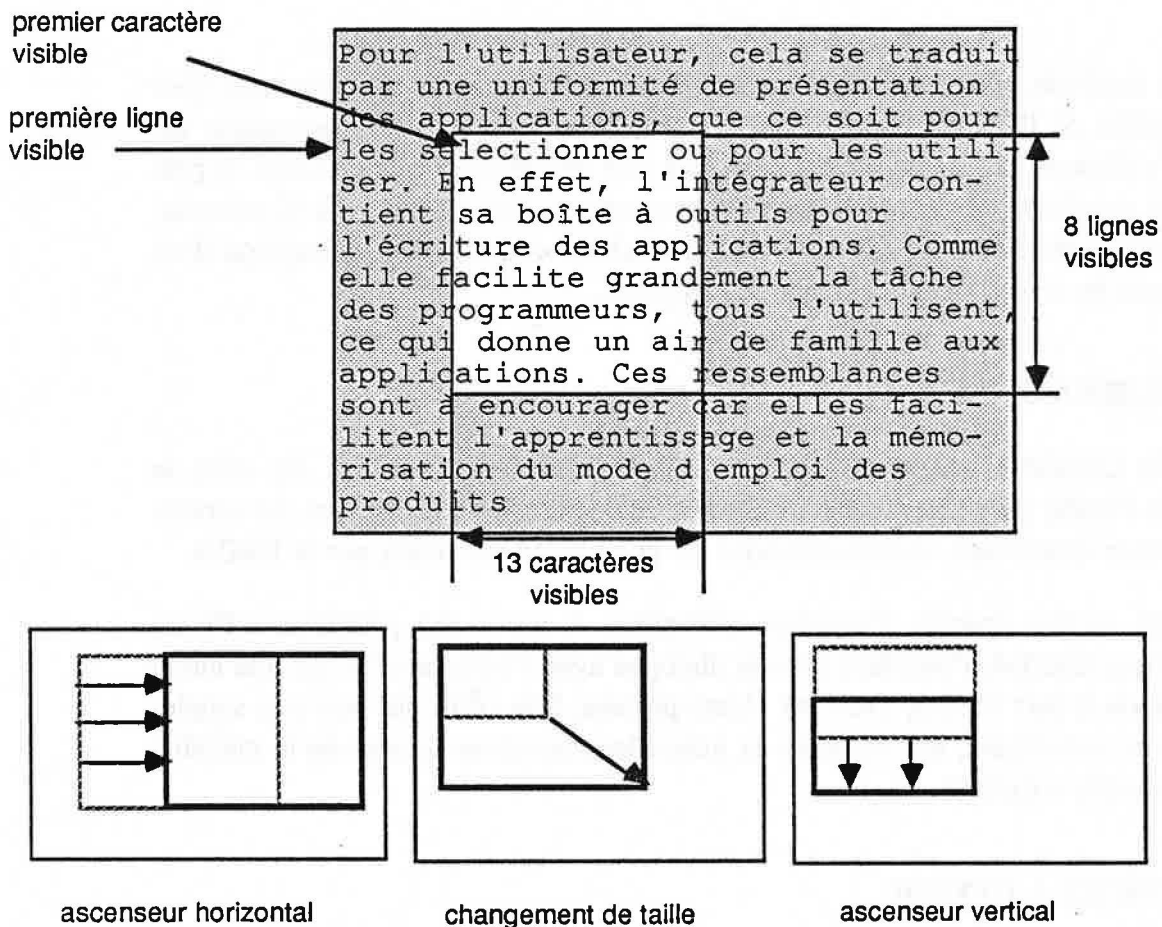


Fig IV.6 : Affichage d'un fichier dans une fenêtre

## 2.11 LES INTEGRATEURS

Un intégrateur est un logiciel qui placé au dessus du système d'exploitation, accueille les applications. Il leur offre les unités logiques décrites précédemment.

Pour l'utilisateur, cela se traduit par une uniformité de présentation des applications, que ce soit pour les sélectionner ou pour les utiliser. En effet, l'intégrateur contient sa boîte à outils pour l'écriture des applications. Comme elle facilite grandement la tâche des programmeurs, tous l'utilisent, ce qui donne un air de famille aux applications. Ces ressemblances sont à encourager car elles facilitent l'apprentissage et la mémorisation des modes d'emploi des produits. Les opérations semblables sont proposées de la même façon et produisent des effets comparables. Les concepteurs doivent respecter une certaine cohérence dans le dialogue entre l'application et l'utilisateur pour ne pas désorienter ce dernier.

Par exemple, pour mettre fin à l'utilisation d'un logiciel, la commande *Quit* est disponible. Si l'état de l'application a changé depuis la dernière sauvegarde sur disque, l'utilisateur doit indiquer s'il désire ou non enregistrer le nouvel état. Il peut également annuler la demande de sortie du programme. Ces efforts dans la présentation des programmes est à rapprocher des travaux en cours dans l'élaboration d'un modèle pour les interfaces utilisateurs [LYNCH86].

### 3 LE BUREAU CRMM

Le modèle CRMM est bien adapté à une spécification de l'interface. En effet, le BBUA est l'entité particulièrement chargée du dialogue avec l'utilisateur. Le service de conférence (archivage, communication) est principalement rendu par le BBGA.

Cette section spécifie l'interface utilisateur. A partir des primitives offertes par le service CRMM, l'interface gère le dialogue avec l'utilisateur. C'est une entité de réalisation à part entière, avec ses objets propres. Elle offre des services supplémentaires et notamment, elle exploite au mieux les ressources locales de la machine sur laquelle elle s'exécute.

#### 3.1 SERVICES A OFFRIR

Pour un usager, les services de base de CRMM sont :

- ajout, marquage, lecture, destruction, effacement de fiches,
- ajout et suppression de liens,
- soumission de fiches à éditeur,
- réponses à demande d'imprimatur,
- lecture des demandes d'imprimatur et des réponse des éditeurs.

Si le modèle défini dans les chapitres précédents décrit le fonctionnement d'une conférence donnée, la réalisation et en particulier l'interface utilisateur considèrent un système de conférences, l'usager a donc accès à un ensemble de conférences. Pour un meilleur usage des services de base du système, certaines facilités sont également prévues.

Seules sont proposées les opérations autorisées. Ainsi, pour un auteur sans les droits d'écrivain, l'ajout de fiches n'est pas proposé. En revanche, il peut soumettre des fiches à un éditeur. Cela matérialise les rôles des membres du groupe.

Un usager peut manipuler un groupe de fiches. Un mécanisme de sélection lui permet de décrire un ensemble de fiches. Par exemple, il demande la destruction des fiches d'un auteur donné.

Certaines requêtes de lecture sont privilégiées, pour les fiches :

- non lues,
- marquées,
- liées à une fiche en cours de lecture,
- qui viennent d'être lues.

### 3.2 STATION DE TRAVAIL

L'accès au serveur CRMM se fait à partir d'un poste de travail personnel. Il s'agit d'exploiter ses ressources propres, il ne doit pas être employé comme simple émulateur de terminal. En particulier, des copies de fiches CRMM pourront être stockées localement. La station dispose d'un écran graphique, d'une souris et d'un système de multifenêtrage (Fig. IV.7).

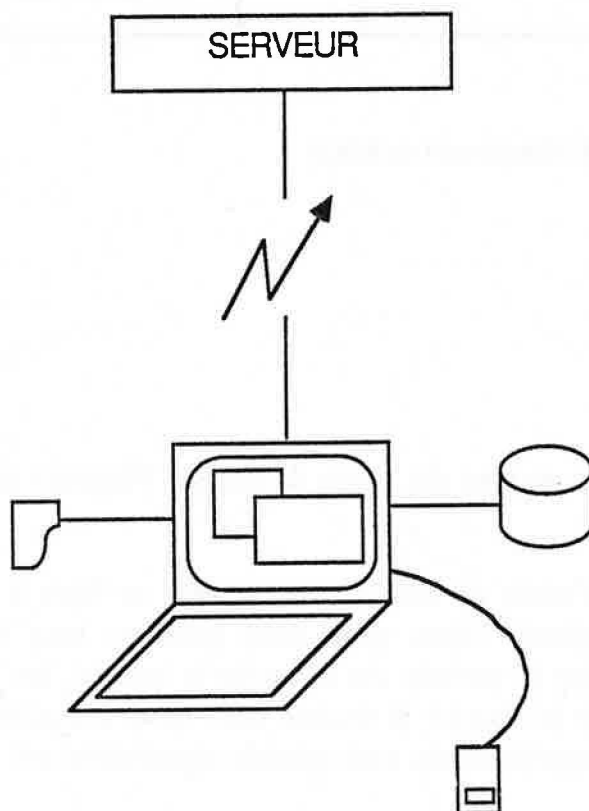


Fig IV.7 : Station de travail

### 3.3 UA ECLATE

Le modèle CRMM doit prendre en compte cet accès à partir de la station de travail. Pour cela, l'entité BBUA qui a en charge l'interface utilisateur est éclatée en deux parties. L'une dite **BBUA client** est déplacée sur la station de travail, l'autre dite **BBUA serveur** demeure sur la machine où le service de conférence est disponible. C'est elle qui représente l'usager auprès du serveur (Fig. IV.8).

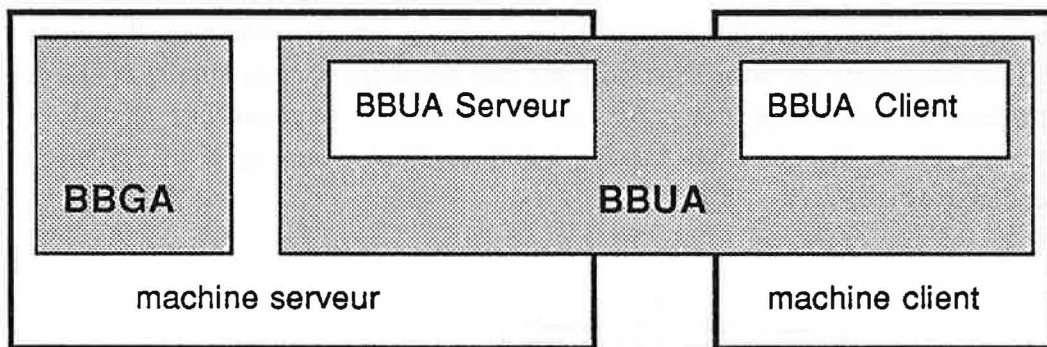


Fig IV.8 : Eclatement du BBUA

Ce découpage demande un partage des objets du BBGA. Plusieurs contraintes influent sur cette répartition.

- Pour un même usager, l'accès au BBUA serveur peut se faire à partir de différentes stations de travail. Donc, entre deux sessions, tous les objets doivent être disponibles sur le serveur. Au début de la session, les objets du BBUA client sont lus par ce dernier. Il restitue ceux qu'il a modifiés en fin de session, ou alors les modifications sont aussitôt répercutées sur l'original détenu par le serveur.

- Le temps d'attente de l'utilisateur doit être minimisé. Pour cela, le maximum d'objets possible doit être disponible sur la machine cliente. En particulier, les fiches qui sont susceptibles d'être lues sont à privilégier, soit en gardant celles qui viennent d'être lues, soit en anticipant celles qui ont de fortes chances d'être demandées par l'utilisateur. Les paragraphes suivants détaillent cette gestion des objets.

### 3.4 LES OBJETS INTERNES DE L'INTERFACE CRMM

Pour rendre les services cités plus haut, l'interface dispose d'objets qui lui sont propres (Fig. IV.9). Cela enrichit l'environnement de l'utilisateur sans alourdir et pénaliser le reste du logiciel. Certains services offerts à l'usager n'impliquent ainsi aucun appel au système. La station de travail dispose de ressources propres. En particulier, elle est pourvue de mémoire qui stocke les objets suivants.

#### 3.4.1 La liste des numéros locaux des fiches non lues,

Cet objet n'est pas directement accessible à l'utilisateur, il permet en revanche de tenir à jour la liste des fiches non lues. Le nombre de ces fiches est fourni dans l'état des conférences. Il est utile pour suivre l'évolution d'une conférence. De plus, cette liste est nécessaire pour offrir un accès rapide et simple aux fiches non encore consultées. En effet, celles-ci sont supposées plus intéressantes pour l'usager.

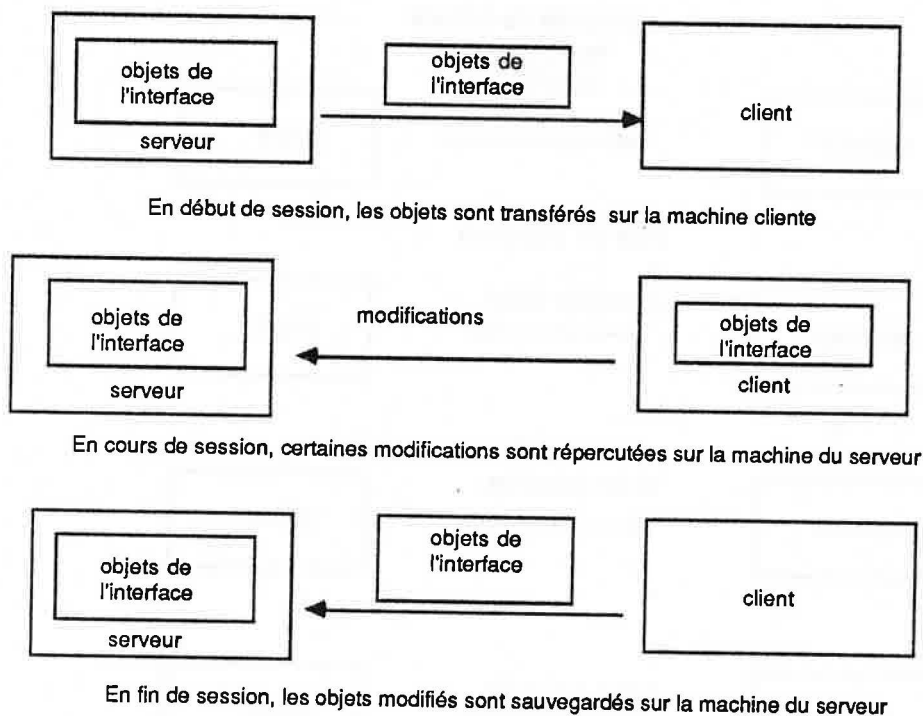


Fig IV.9 : Partage des objets entre client et serveur

### 3.4.2 La liste des fiches précédentes

Elle permet à l'utilisateur de redemander les fiches lues depuis le début de la session. Cela prévient toute attente pour un objet qui vient juste d'être lu, en effet, ce type de contrainte paraîtrait incompréhensible pour l'utilisateur. Dans son univers, l'objet semble proche (dans le temps), une recherche trop longue révélerait un outil primitif.

Cette liste contient les numéros locaux des n dernières fiches lues pendant la session en cours. Une fiche est considérée comme lue, si elle a été affichée dans une fenêtre et si cette fenêtre a été fermée.

### 3.4.3 La liste des fiches sélectionnées

La plupart des opérations sur la base des fiches portent sur un ensemble de fiches décrit par une chaîne de sélection. Lors d'une demande de lecture, il est préférable de ne pas transmettre systématiquement toutes les fiches qui correspondent à la requête. Après une demande d'opération qui met en jeu une sélection, la base rend la liste des numéros de fiches valides, cette liste est transmise au poste de travail. Dans le cas d'une lecture, la première fiche est jointe à la liste puis les demandes des fiches suivantes se font à l'unité (Fig. IV.10). Toutefois, il est possible d'anticiper les demandes de l'utilisateur.

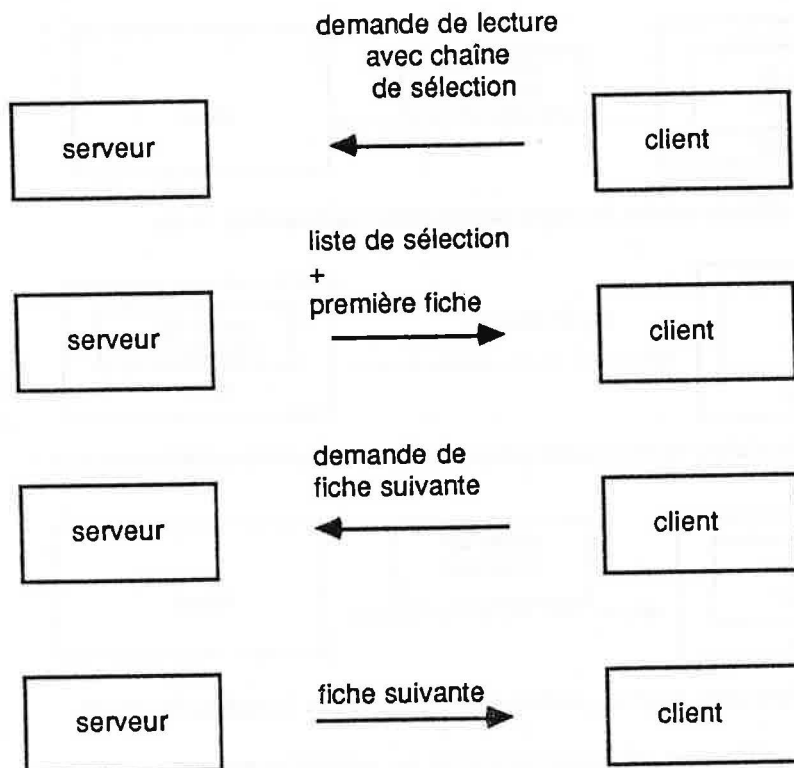


Fig IV.10 : Mécanisme de lecture

#### 3.4.4 La liste des fiches marquées

Elle est chargée en début de session, puis est sauvegardée en fin de travail. Toute la gestion de cette liste pendant la session est à la charge de l'interface. Les modifications (ajout, suppression de marques) demeurent proprement locales et n'entraînent pas d'appel au serveur. Chaque référence à une fiche marquée est transformée, le numéro local remplace la valeur de la marque.

#### 3.4.5 La liste des alias des sites

Un alias permet de désigner un site. Cette forme est plus pratique et plus évocatrice qu'un nom d'O/R. L'alias sert à la désignation des usagers et des fiches. La liste de ces noms est fournie à la machine cliente au début de la session.

#### 3.4.6 Le contenu de la boîte

La liste des messages contenus dans la boîte est transférée sur le poste de travail au début de la session. Les messages ont une double identification, une interne au système et une externe pour l'utilisateur. Cette seconde doit être plus représentative du contenu, elle est formée à partir du nom de l'expéditeur. Les demandes de lecture se font à l'unité, les messages transférés sont stockés localement par le système pour une éventuelle relecture, ceci toujours dans le but de minimiser les échanges. Les modifications de la boîte (demande de destruction de messages) sont immédiatement répercutées sur le serveur, la vision locale de la boîte n'est pas sauvegardée en fin de session (Fig. IV.11). Les opérations possibles sur la boîte dépendent évidemment du rôle de l'utilisateur.

- un utilisateur éditeur reçoit des demandes d'imprimatur, il les lit et y répond. Cette dernière opération détruit le message.
- un utilisateur non écrivain reçoit des réponses d'imprimatur, il les lit et les détruit.
- Les autres utilisateurs n'ont pas de boîtes aux lettres pour CRMM.



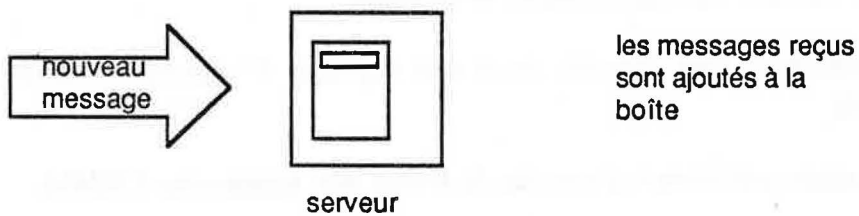
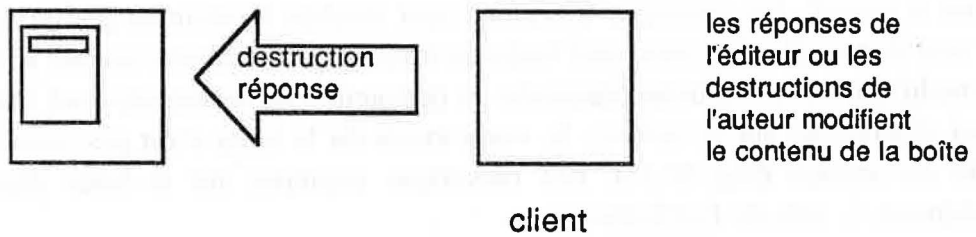
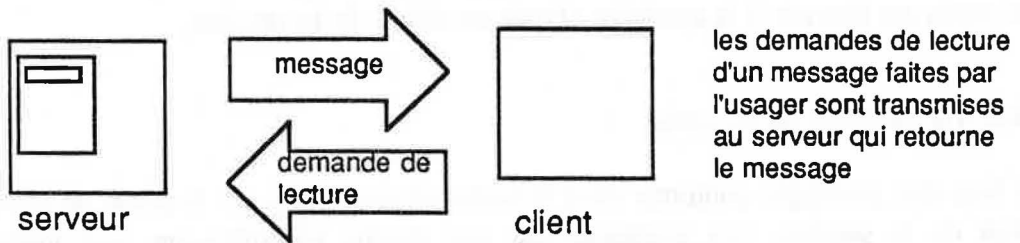
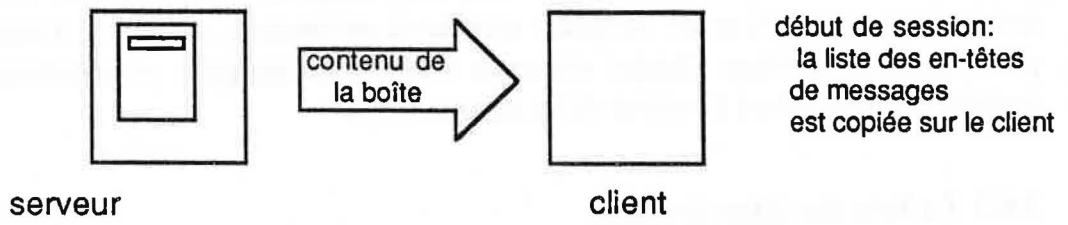


Fig IV.11 : Gestion de la boîte aux lettres

### 3.4.7 La description de la conférence

La description de la conférence est importante pour l'utilisateur, elle comprend les règles d'utilisation de la conférence, le savoir-vivre indispensable à tout travail en groupe. En plus des objets donnés ci-dessous, un mode d'emploi en ligne est disponible.

**Le nom de la conférence** Ce nom sert pour désigner la conférence dans le corps des fiches ou extérieurement au système. La raison d'existence de la conférence est aussi donnée, ainsi qu'une description du public qu'elle concerne.

**Les droits de l'utilisateur** Ils fixent les opérations autorisées à l'usager. Ils précisent les rôles qu'il peut jouer. A la connexion dans une conférence donnée, les droits de l'utilisateur sont transmis. L'interface en tient compte pour n'offrir que les opérations légales.

**Les liens disponibles** Il s'agit des relations qui peuvent exister entre fiches, ils varient d'une conférence à l'autre. Pour chacun d'eux, le sens et le comportement est expliqué.

### 3.4.8 La description de l'ensemble des conférences

Ce document décrit le fonctionnement d'une conférence. Il est néanmoins évident qu'un même site peut abriter plusieurs conférences (un BBGA pour chacune d'elles). L'interface permet l'accès à toutes celles auxquelles l'usager est abonné. Lors de la connexion, ou à la demande de l'usager, il est possible de changer de conférence. A tout instant, un état est disponible. Pour chaque conférence à laquelle l'usager est abonné, l'interface donne le nom de la conférence et le nombre de fiches non lues.

### 3.4.9 Les fiches disponibles localement

Pour limiter l'attente de l'utilisateur pendant les communications avec le serveur, l'interface prévoit de stocker des fiches sur la machine client. Cette mémorisation peut se faire par anticipation, ainsi, puisque la liste des fiches non lues est connue, il est possible de devancer les demandes probables de l'utilisateur, et de transférer les fiches de cette liste sur le poste de travail. Les fiches de la liste de sélection peuvent elles-aussi être transférées par anticipation. Il est possible également de conserver les fiches qui viennent d'être lues, en supposant que le lecteur puisse en redemander l'affichage.

### 3.5 LES OBJETS D’AFFICHAGE

L’application manipule des fenêtres pour dialoguer avec l’usager. Ces fenêtres sont graphiques. Elles apparaissent dans une fenêtre principale qui limite la part d’écran qui est dévolue à l’application CRMM.

#### 3.5.1 La fenêtre de lecture

Une fenêtre de lecture (Fig. IV.12) contient une fiche, l’en-tête est optionnellement visible. La partie lisible est déplaçable par ascenseurs (horizontal et vertical) et sa taille peut varier. Le remplissage est optimisé, la taille est limitée au maximum nécessaire. Il est possible que plusieurs de ces fenêtres soient présentes simultanément à l’écran (Fig. IV.13).

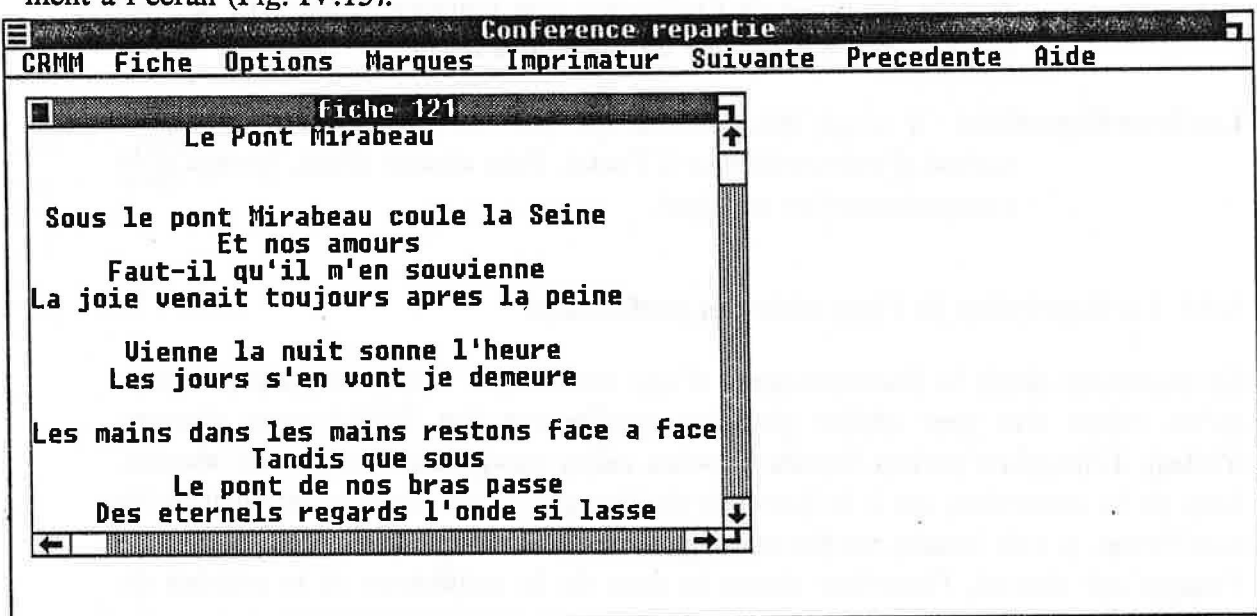
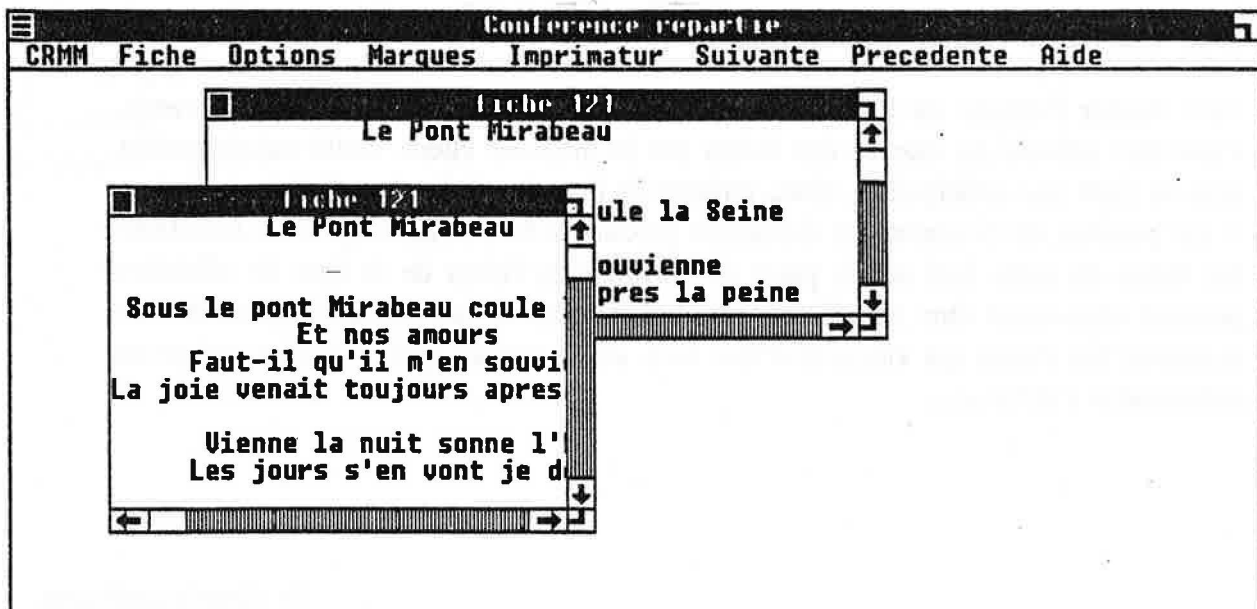


Fig IV.12 : Fenêtre de lecture



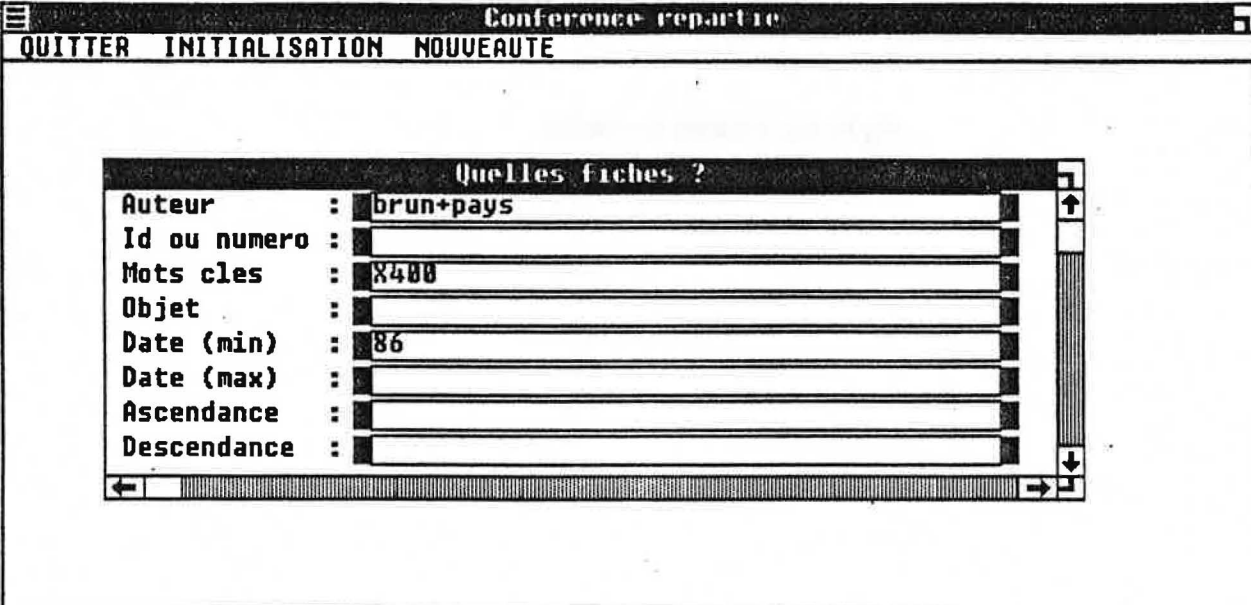
### 3.5.2 La fenêtre de sélection de fiches

La fenêtre de sélection permet la saisie de requêtes par forme (QBF : Query By Form). Les critères autorisés sont :

- identifiant général,
- numéro local,
- date de début de recherche,
- date de fin de recherche,
- auteur,
- mots clés,
- objets,
- lien ascendant,
- lien descendant.

Entre eux, il existe un ET implicite. A l'intérieur d'un critère, plusieurs valeurs sont autorisées (sauf pour les dates), il existe un OU entre elles (Fig. IV.14). Cette solution ne permet pas toutes les possibilités de sélection, une autre fenêtre permet de saisir directement une requête sous forme algébrique.

L'auteur peut également affiner son choix en indiquant des préférences grâce au menu de sélection décrit plus loin.



The screenshot shows a window titled "Conférence repartie" with a menu bar containing "QUITTER", "INITIALISATION", and "NOUVEAUTE". Inside the window, a dialog box titled "Quelles fiches ?" is open. The dialog box contains the following fields:

Auteur	:	brun+pays
Id ou numero	:	
Mots clés	:	8488
Objet	:	
Date (min)	:	86
Date (max)	:	
Ascendance	:	
Descendance	:	

Fig IV.14 : Fenêtre de sélection

### 3.5.3 La fenêtre de résultat

La fenêtre apparaît sur l'écran pour une lecture, elle ne peut ni être déplacée, ni agrandie, ni diminuée. Elle permet de copier les résultats dans un fichier du système d'exploitation de la machine cliente. (Fig. IV.15).

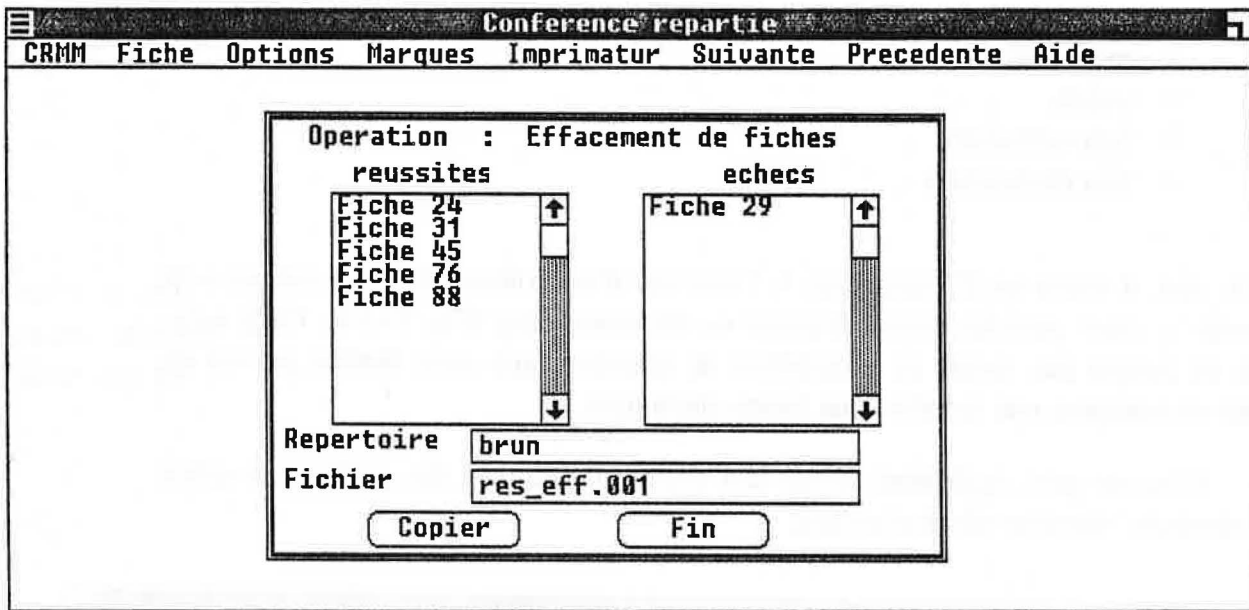


Fig IV.15 : Fenêtre de résultat

### 3.5.4 La fenêtre des marques

La fenêtre des marques (Fig. IV.16) permet de sélectionner une fiche marquée pour la lire, d'ajouter ou de supprimer une marque.

### 3.5.5 La fenêtre de la boîte aux lettres

Cette fenêtre liste le contenu des messages. Elle permet d'en demander la lecture et selon le rôle de l'utilisateur autorise la réponse (éditeur) ou la destruction (auteur). (Fig. IV.17).

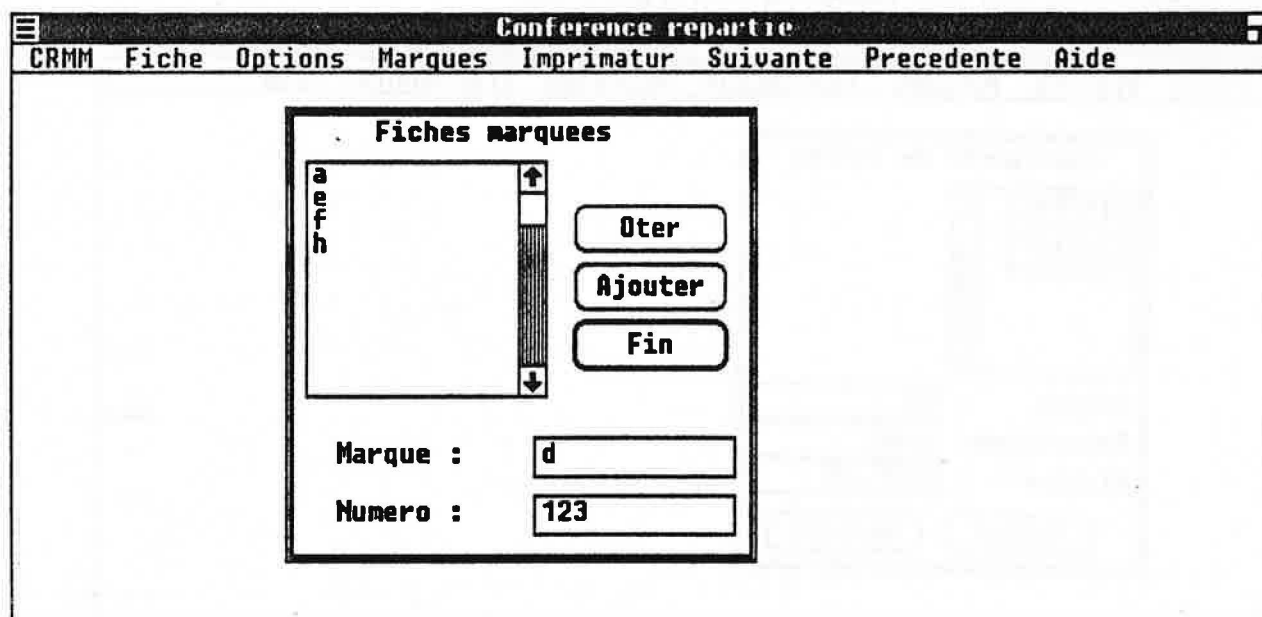


Fig IV.16 : Fenêtre des marques

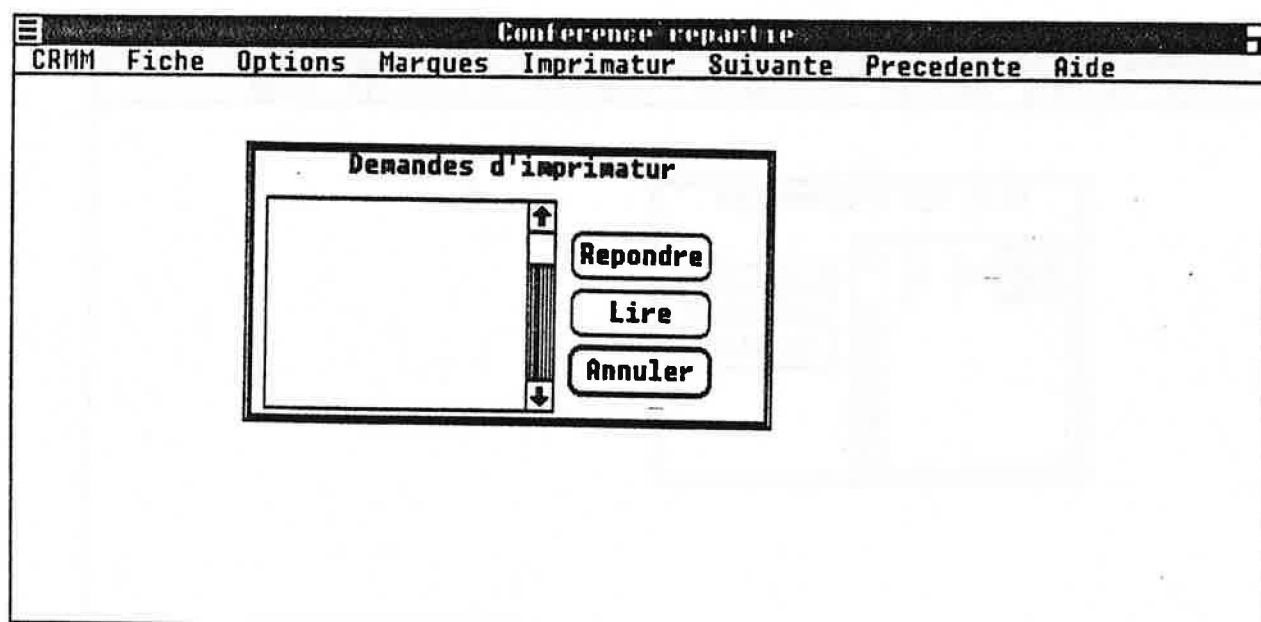


Fig IV.17 : Fenêtre de la boîte aux lettres

### 3.5.6 La fenêtre de copie

La fenêtre de copie saisit le dialogue (désignation de la fiche, du répertoire d'accueil, du fichier copié) nécessaire lorsque l'utilisateur désire copier une fiche dans un fichier du système d'exploitation de la machine cliente (Fig. IV.18).

### 3.5.7 La fenêtre d'état

La fenêtre d'état liste les conférences auxquelles l'utilisateur est abonné et donne le nombre de fiches non lues pour chacune d'elles (Fig. IV.19).

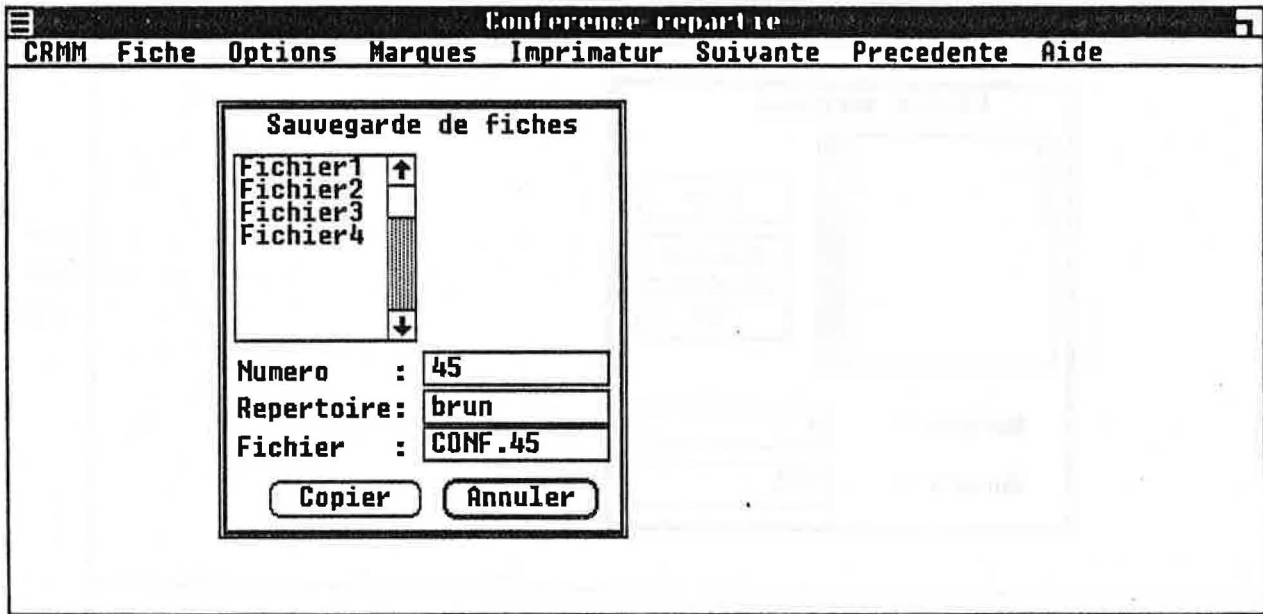


Fig IV.18 : Fenêtre de copie

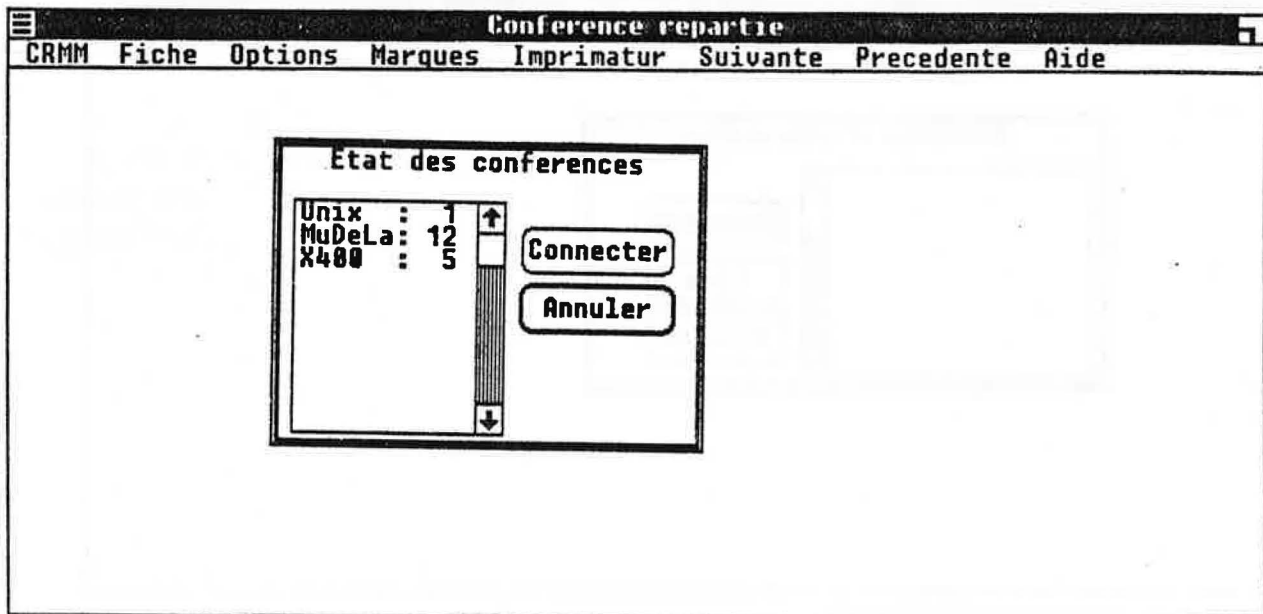


Fig IV.19 : Fenêtre d'état des conférences

### 3.5.8 La fenêtre de création de fiche

Dans cette fenêtre, l'utilisateur crée une fiche. Il donne les attributs qui ne peuvent être générés automatiquement : objets, mots clés et durée de validité de la fiche. A part l'objet, ils sont facultatifs.

Ensuite, il a le choix entre indiquer le nom du fichier où se trouve le texte de la fiche et utiliser l'éditeur de texte inclus dans la fenêtre.

Objet	Conférence répartie
Mots clés	X400
Lien	reponse:23
Validité	juin88
Fichier	

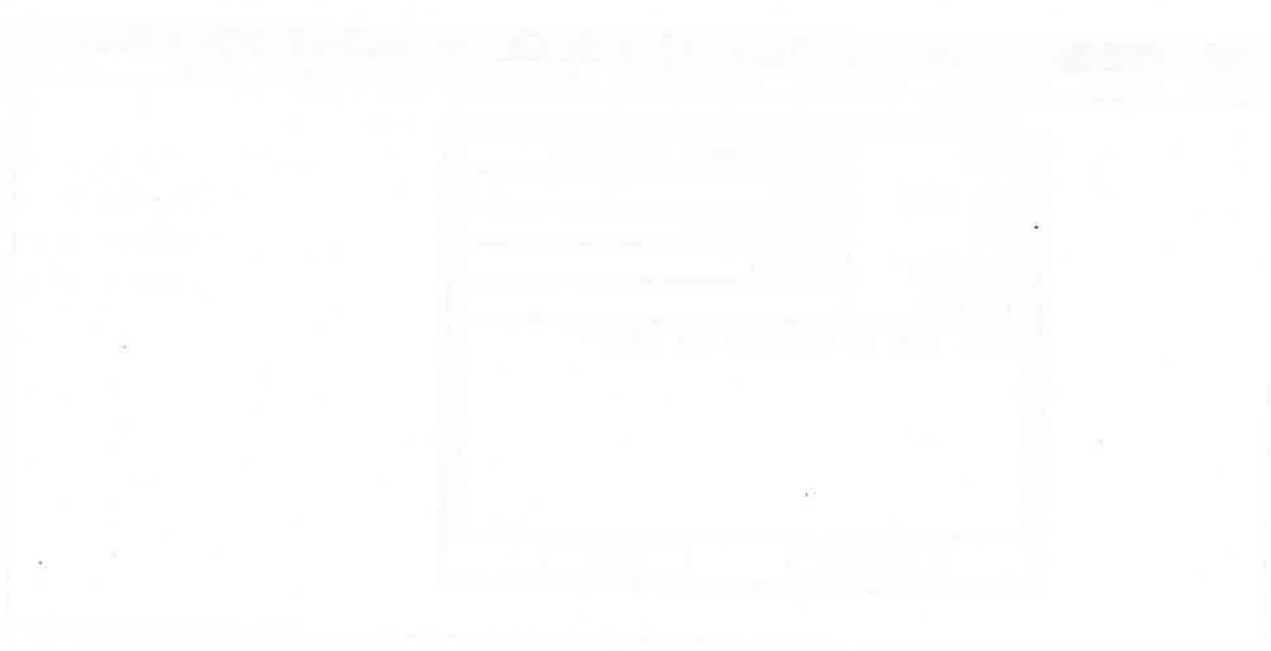
Ceci est un exemple de fiche

Annuler Fin

Fig IV.20 : Fenêtre de création de fiche



... de l'interface utilisateur ...



## **REALISATION**

**113**

---

<b>LES OUTILS UTILISES</b>	<b>113</b>
Windows	113
Le toolkit de développement	114
GDI	115
Windows	115
Les fenêtres de Windows.	116
Fonctionnement de l'intégrateur	117
Evénements et messages	117
Fonction principale	118
Fonction de traitement des messages	118
Fenêtres prédéfinies	122
Boîtes de dialogue	122
Boîte de messages	124
Les menus	125
Les ressources	125
Réalisation d'une application windows	125
Gestion du multifenêtrage	126
<b>PRESENTATION GENERALE DES MODULES DE CRMM</b>	<b>131</b>
<b>ECLATEMENT DU MODULE IU</b>	<b>132</b>
Communication client-serveur	132
Les objets IU serveur	133
Les objets IU client	133
Les fenêtres	135
Les menus	137



## Chapitre 5

### REALISATION

Ce chapitre présente d'abord les outils choisis pour réaliser CRMM. Il insiste sur l'intégrateur Windows de Microsoft. Il ne reprend pas la documentation, mais fournit des informations essentielles sur la mise en œuvre d'applications, particulièrement celles que j'aurais aimé trouver dans la documentation au début de la réalisation. Il décrit succinctement les principaux modules de CRMM. Ensuite, il présente les objets et modules propres à l'interface utilisateur.

#### 1 LES OUTILS UTILISES

Le poste de travail choisi est un compatible PC-AT qui tourne sous MS-DOS 3.1. Il est équipé d'une souris et d'un écran graphique. Les logiciels d'interface utilisateur sont écrits en C, et ils sont prévus pour être intégrés dans MS-Windows. Le serveur est un SPS 7 de BULL qui tourne sous UNIX système V, les logiciels sont écrits en C.

Le choix d'UNIX s'explique par l'appartenance de CRMM au projet SMARTIX, C devient alors évident (ces raisons rencontrent opportunément les préférences de l'auteur). Le choix du PC-AT et de Windows est dicté par la large diffusion de ces produits plus que par leurs qualités intrinsèques.

##### 1.1 WINDOWS

Ce paragraphe donne une rapide description de Windows. Elle est nécessaire pour la compréhension de la réalisation, pour plus d'informations sur ce produit, il faut lire la documentation de Microsoft [MICROSOFT87].

Windows est un intégrateur d'applications de Microsoft. Il fonctionne sur les ordinateurs individuels compatibles avec les micro-ordinateurs IBM. Il est prévu pour tourner sur le DOS de Microsoft. Il gère des fenêtres graphiques, une souris et des menus déroulants. Les applications "bien écrites" (*well behaved*) dialoguent exclusivement avec Windows, notamment pour les entrées/sorties (ports, mémoire d'écran, clavier). Dans ces conditions, l'écran peut afficher plusieurs applications simultanément. Elles tournent dans des fenêtres. Les fenêtres appartenant à des applications différentes ne se recouvrent pas, elles partitionnent l'écran physique. En revanche, à l'intérieur de sa fenêtre principale, l'application peut posséder des fenêtres qui se superposent.

Pour ces applications bien écrites, l'intégrateur donne toute sa mesure. Le fonctionnement semble multitâche. Toutefois, la suite du chapitre montre que le partage du temps entre applications n'est pas digne de cette appellation.

## 1.2 LE TOOLKIT DE DEVELOPPEMENT

Microsoft fournit un *toolkit* qui permet le développement d'applications intégrables. Il s'agit d'une boîte à outils dont le chapitre précédent définit la notion. Elle est principalement composée des bibliothèques de fonctions suivantes (Fig. V.1).

- une fournit des appels aux ressources du système (mémoire, modules, tâches, fichiers).
- la seconde (GDI) offre des fonctions graphiques,
- la troisième manipule des objets de plus haut niveau, comme les fenêtres, les menus. C'est la bibliothèque Windows proprement dite.

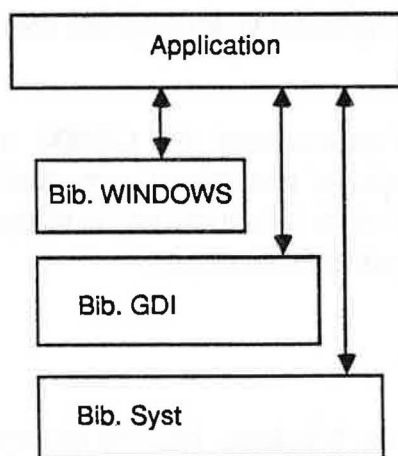


Fig V.1 : Appels aux bibliothèques par une application Windows

### 1.2.1 GDI

GDI (*Graphic Device Interface*) offre une vue logique de périphériques graphiques tels que l'écran. A ce titre, il s'apparente à des produits comme GKS ou VDI<sup>1</sup>. Il se charge de traduire les appels de l'application en primitives compréhensibles par le périphérique. Les fonctions manipulent des objets graphiques élémentaires comme les segments de droite, les cercles ou les rectangles. Elles permettent de les tracer avec certains attributs, comme la couleur, le fond...

A chaque fonction graphique, est associé un contexte d'affichage (*Display context*). C'est un ensemble de données utilisées et gérées par GDI. Il décrit un périphérique graphique et son pilote (*driver*). Parmi les informations qu'il contient, on peut citer à titre d'exemple :

- la couleur de la brosse pour peindre,
- la couleur du crayon pour écrire,
- la police de caractères,
- la position courante du crayon,
- la couleur du texte,
- la table des couleurs disponibles,
- ...

### 1.2.2 Windows

Les fonctions offertes au niveau Windows manipulent les objets plus abstraits comme les fenêtres avec des opérations comme la création, l'ouverture, la fermeture, le déplacement, le changement de taille... Elles travaillent également sur les objets menus, en permettant leur modification (ajout, destruction, changement, inhibition, validation de choix) et leur suppression.

---

1 VDI : Virtual Device Interface, il est utilisé par l'intégrateur multifenêtre GEM de Digital Research.

### 1.3 LES FENETRES DE WINDOWS.

Les fenêtres de Windows sont des objets décrits par :

- un type :
  - *tiled window* : ces fenêtres partitionnent l'écran physique sans se recouvrir, comme un carrelage. La fenêtre principale de l'application est de ce type.
  - *child window* : ces fenêtres "filles" sont à l'intérieur d'une autre fenêtre. Elles peuvent se recouvrir.
  - *popup window* : Ces fenêtres apparaissent au premier plan de l'écran, le temps d'un message, ou d'une question.
- des dimensions, largeur et hauteur,
- une position par rapport à la fenêtre qui les contient, ou par rapport à l'écran physique,
- des attributs de présentation (couleur, cadre...),
- éventuellement un ascenseur horizontal et un vertical,
- un titre,
- une boîte de taille (*size box*), qui permet d'agrandir ou de diminuer une fenêtre,
- une barre de déplacement,
- une icône qui peut éventuellement représenter la fenêtre lorsque celle-ci est fermée,
- un menu déroulant spécifique à la fenêtre,
- un menu dit "système" pour les options universelles : fermeture, zoom ou déplacement,
- une fonction qui traite les messages concernant la fenêtre,
- un identifiant unique.

## 1.4 FONCTIONNEMENT DE L'INTEGRATEUR

### 1.4.1 Evénements et messages

Les actions de l'utilisateur sont perçues par le système comme des événements, par exemple :

- déplacement de la souris,
- click de la souris dans une fenêtre,
- frappe d'une touche du clavier,
- manipulation de fenêtres (taille, position),
- click sur un choix de menu.

Pour chaque application, Windows gère une file d'attente. Lorsqu'il détecte un événement, il dépose un message le décrivant dans la file de l'application concernée.

Un message contient :

- l'identifiant de la fenêtre destinatrice du message,
- l'identifiant du message,
- deux paramètres dépendant du message,
- la date de l'événement cause du message,
- position de la souris au moment de l'événement.

Par exemple, si l'utilisateur clique dans une fenêtre, un message le signale à l'application détentrice de cette fenêtre.



### 1.4.2 Fonction principale

La fonction principale d'une application doit lire la file des messages et les diriger vers les fonctions associées :

**répéter**

**lire** message dans la file;  
transmettre le message à la fonction concernée;

**jusqu'à** la fin de l'application;

La fin de l'application est repérée par un message particulier.

Tant que la fonction n'a pas terminé le traitement d'un message donné, le système ne peut scruter les files d'attente des autres applications qui sont de ce fait bloquées. Windows permet un partage des ressources, mais l'accès n'est pas géré efficacement pour que le système puisse être dit multitâche.

### 1.4.3 Fonction de traitement des messages

Le programmeur associe une fonction à chaque fenêtre. Elle traite les messages qui lui sont destinés. Sa structure est la suivante.

**selon** type du message **faire**

type 1 :...  
type 2 :...  
**defaut** : DefWindowProc()

**fin du selon**

Lorsque l'application désire ne pas traiter un message, la fonction *DefWindowProc()* procède au traitement par défaut prévu par le système.

La bonne connaissance du fonctionnement du système Windows exige la compréhension de ce qui est automatiquement géré et de ce qui reste à la charge du logiciel d'application. Reprenons la gestion d'une fenêtre qui affiche le contenu d'un fichier et voyons selon les événements, les messages qu'il faut traiter.

**WM\_PAINT** : Il arrive que des parties de fenêtres ne soient plus à jour. Cela provient soit de changement de la forme de la fenêtre, ouverture, fermeture, changement de taille ou de position, soit de l'évolution de son contenu.

Dans le premier cas, le système peut détecter les changements, et calculer les zones de l'écran à mettre à jour. Ces régions sont alors invalidées à l'initiative du système. Dans le deuxième cas, les modifications dépendent de l'application, il revient donc à cette dernière de signaler au système les zones concernées.

Lorsqu'une région est à mettre jour, le système envoie à l'application le message **WM\_PAINT** dont les paramètres repèrent la région invalide.

La fonction qui traite ce message peut alors repeindre entièrement la fenêtre, ou alors pour gagner du temps, limiter le traitement à la zone décrite par les paramètres. De toutes façons, le système masque les régions à jour ou invisibles. Lorsque la fonction a achevé de repeindre la fenêtre, elle le signale au système. Ce dernier considère alors que la région est à jour.

Dans l'exemple étudié, la fenêtre est systématiquement repeinte, le temps de calcul est suffisamment court pour que cela soit tolérable. Le traitement est alors :

**début**

```

BeginPaint()  -- indication de début de mise à jour
pl ← numéro de la première ligne visible
nl ← nombre de lignes visibles
pc ← ordre du premier caractère visible
nc ← nombre de caractères visibles
pour i := pl à pl + nl faire
  début
  l ← ligne i du fichier
  nce ← Min( nc, longueur(l) - pc )
  pour j ← pc à pc + nce afficher le caractère j de l
  fin
EndPaint()    -- indication de fin de mise à jour
fin

```

**WM\_VSCROLL** : Ce message arrive pour chaque manipulation de l'ascenseur vertical. L'utilisateur demande de décaler le texte vers le haut ou le bas d'une ligne ou d'une page, d'afficher le début ou la fin du texte, et d'aller directement à une ligne du texte. Le programme contrôle la course de l'ascenseur en fonction du nombre de lignes visibles, de la première ligne visible et du nombre de lignes du fichier.

Une fois que ces variables d'état du système sont à jour, il faut effectivement repeindre la fenêtre. Pour cela, il existe deux possibilités. La première est la plus simple, il suffit d'invalider la fenêtre entière par un appel à *InvalidateRect()* en indiquant que la totalité de la fenêtre est concernée. La prochaine réception de WM\_PAINT provoquera la restauration de la fenêtre entière.

La seconde est méthode est plus fine, elle consiste à peindre immédiatement les fenêtres sans utiliser les régions valides et le clipping du système. Prenons l'exemple d'un déplacement du texte d'une ligne vers le bas à l'intérieur de la fenêtre. Pour obtenir cet effet, il suffit de déplacer le texte vers le bas d'une hauteur d'une ligne et de n'afficher que la nouvelle ligne qui sommet de la fenêtre. Le déplacement du texte est réalisé par un mouvement de la bitmap de l'écran. Cette dernière méthode est plus élégante, mais est coûteuse en lignes de code. Il faut calculer les zones à déplacer pour tous les types d'opération. De plus, la régénération totale et systématique de la fenêtre n'est pas trop pénalisante en temps d'exécution, c'est pourquoi elle est suffisante dans notre exemple.

**WM\_HSCROLL** : ce message reflète une action sur l'ascenseur horizontal. Les traitements sont comparables aux précédents, sauf que les mouvements de texte se font horizontalement.

**WM\_SIZED** : il est reçu chaque fois que la taille de la fenêtre change, ses paramètres donnent les nouvelles dimensions. L'application doit calculer le nombre de lignes et le nombre de caractères visibles. Si la fenêtre est trop grande par rapport au texte à visualiser, elle doit être retaillée. De plus, il faut optimiser le remplissage de la fenêtre pour que le maximum de texte soit visible.

## 1.5 FENETRES PREDEFINIES

Ce qui précède montre que la majeure partie de la gestion de l'affichage est à la charge du programmeur. Dans le chapitre concernant l'interface utilisateur, la notion d'entrées/sorties de haut niveau a été introduite. Sans répondre entièrement à ce problème, le système Windows propose certaines facilités, à travers des fenêtres prédéfinies qui sont d'usage courant. Leur gestion est en grande partie assurée par le système.

### 1.5.1 Boîtes de dialogue

Ce sont des fenêtres qui permettent le dialogue avec l'utilisateur : affichage, saisie de texte, choix dans une liste de propositions, confirmation, annulation, validation...

L'exemple suivant utilise une telle boîte. Elle comprend la plupart des types de contrôles proposés par Windows (Fig. V.2).

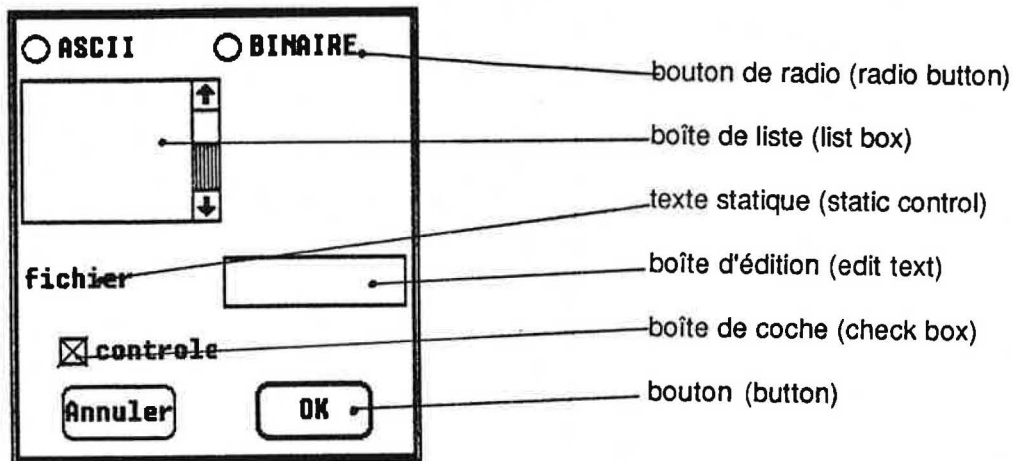


Fig V.2 : Boîte de dialogue de Windows

#### bouton de radio

Ils permettent de faire un choix exclusif.

#### boîte de coche

L'usager choisit ou non une option.

La boîte est décrite dans le fichier des ressources par le script suivant (Fig. V.3). Chaque contrôle est identifié par une constante. Il est repéré par ses coordonnées par rapport au coin supérieur gauche de la boîte. La position de cette dernière est donnée relativement à sa fenêtre parente.

```
SELBOX DIALOG 20, 5, 100, 150
STYLE WS_DLGFRAE|WS_POPUP
BEGIN
RADIOBUTTON "ASCII", -1, 1,1,40,17,MAB
RADIOBUTTON "BINAIRE", -1, 50, 1,40, 17, MAB
LISTBOX -1, 1,21, 50, 50, MAB
LTEXT "fichier", -1, 1, 85, 50, 17
EDITTEXT SELAUT, 52, 82,47,17,MAB
CHECKBOX "controle :", -1, 10, 105, 40, 17
PUSHBUTTON "Annuler", SELCANCEL, 10, 125, 30, 20, MAB
DEFPUSHBUTTON "OK", SELOK, 60, 125, 30, 20, MAB
END
```

Fig V.3 : Description de la boîte de dialogue

### boîte de liste

Ces boîtes (*list box*) affichent une liste de choix. L'utilisateur peut sélectionner l'un d'entre eux en cliquant avec la souris. L'ascenseur vertical permet de gérer des listes suffisamment longues. Ces boîtes sont souvent utilisées pour choisir un fichier dans une répertoire.

Le logiciel d'application peut :

- ajouter un item,
- détruire un ou tous les choix,
- sélectionner un choix,
- demander la sélection courante,
- demander l'état d'un choix,
- ajouter le contenu d'un répertoire,
- ..

Il reçoit un message pour les événements suivants :

- changement de le choix sélectionné,
- double click d'un choix.

**boîte d'édition**

Dans cette boîte (*edit box*) s'exécute un éditeur de texte minimal qui gère un curseur et interprète la frappe de touches du clavier, qu'elles soient alphanumériques ou spéciales (effacement arrière). La position du curseur peut être déplacée au moyen de la souris.

Le logiciel peut entre autres possibilités, lire ou écrire le contenu du contrôle, manipuler le cadre qui englobe le texte, déplacer (*scroll*) le contenu du contrôle. Il est prévenu par messages de la modification du contrôle (contenu, apparence).

**boutons**

Il s'agit d'un cadre contenant un texte. Ils sont sensibles au simple et au double click. L'application en est avertie.

**textes statiques**

Ils permettent l'affichage simple de chaînes de caractères

**1.5.2 Boîte de messages**

Une telle boîte (*message box*) est une *popup window* avec un titre, un message explicatif et des boutons optionnels. Elle est mise en œuvre par un simple appel à la fonction Windows *MessageBox()*.

## 1.6 LES MENUS

Les menus de windows apparaissent au sommet de la fenêtre de l'application. Ils sont déroulants. Ils sont décrits dans le fichier de ressources où chaque choix est défini et identifié. Lorsque l'utilisateur clique l'un d'eux, la fonction qui gère la fenêtre du menu est avertie et la valeur de l'identifiant lui est fournie.

## 1.7 LES RESSOURCES

Les ressources attachées à une application comprennent :

- les menus,
- les icônes,
- les boîtes de dialogues,
- les chaînes de caractères utilisées par le programme.

Elles sont décrites dans un fichier qui est compilé puis lié à l'exécutable.

## 1.8 REALISATION D'UNE APPLICATION WINDOWS

L'écriture d'un programme intégré comprend donc les phases suivantes.

- description des ressources,
- initialisation du programme :
  - création des fenêtres,
  - déclaration et initialisation des structures de données,
- fonction principale,
- fonctions de chaque fenêtre ou boîte de dialogue.



## 1.9 GESTION DU MULTIFENETRAGE

La mise en œuvre d'applications sous Windows n'est pas une tâche facile. Un point particulièrement difficile concerne le recouvrement des fenêtres. Il est très peu étudié dans la documentation, il est donc utile pour les futurs développeurs de décrire comment les fenêtres sont gérées dans le système CRMM.

Lorsqu'une fenêtre n'est plus à jour, la fonction qui la gère reçoit le message WM\_PAINT. Ses paramètres repèrent la zone à repeindre. Malheureusement, il s'agit en fait de la boîte rectangulaire, englobant la zone incriminée. D'autres intégrateurs décrivent une zone par un ensemble de rectangles. Cette option, plus exacte, se révèle plus intéressante pour une bonne gestion du recouvrement des fenêtres (Fig. V.4).

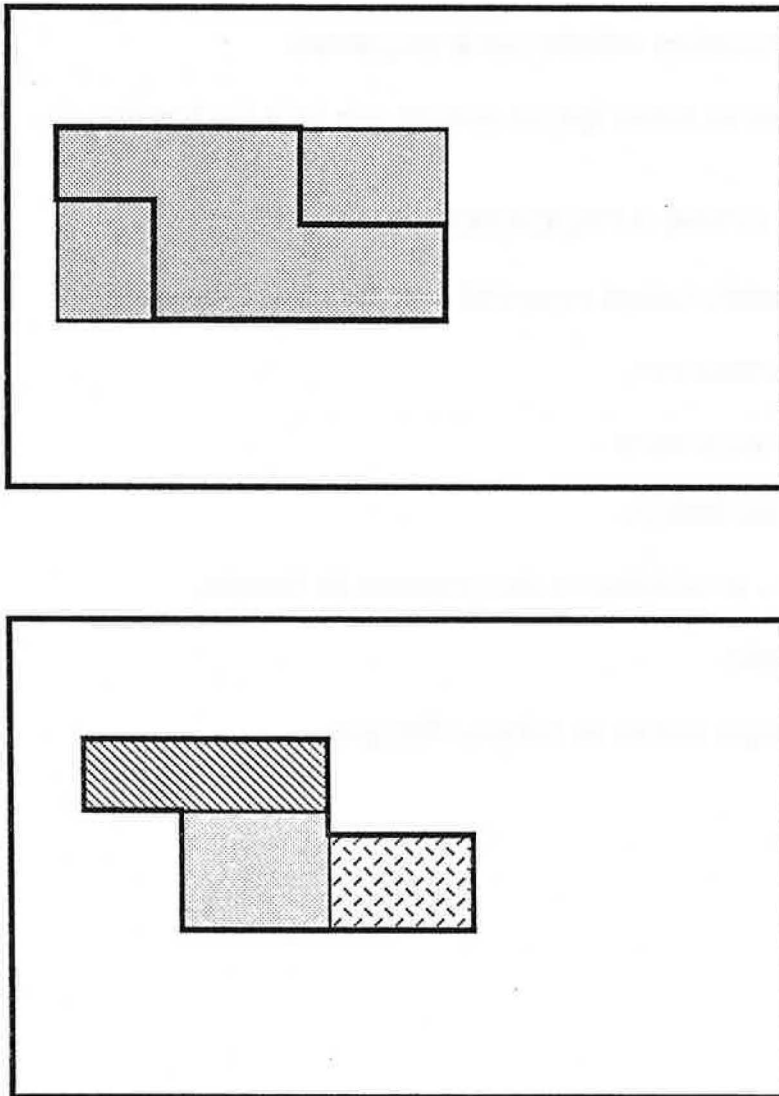


Fig V. 4 : Description de régions par boîte englobante ou par liste de rectangles

Les fenêtres de Windows ont la réputation de ne pas pouvoir se recouvrir. Cela est vrai pour les fenêtres principales des applications. Elles partitionnent l'écran. En revanche, ces mêmes fenêtres peuvent abriter des *child windows* susceptibles de se superposer. Il revient toutefois à l'application de prendre certaines précautions.

En effet, Windows n'assure pas une gestion suffisante de ces *child windows*. Soit une fenêtre principale, abritant trois fenêtres filles (Fig. V.5).

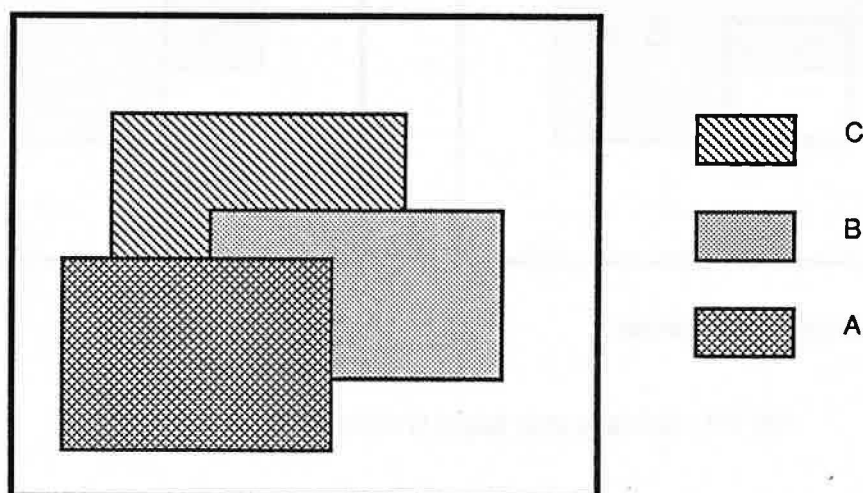
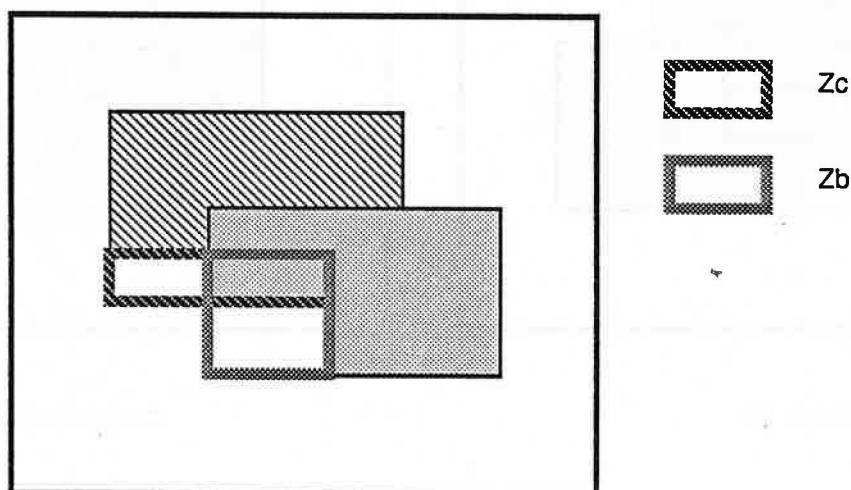


Fig V.5 : Une fenêtre avec 3 child windows

Lorsque l'application ferme la fenêtre A, le système rend invalides pour les fenêtres B et C, les zones Z<sub>b</sub> et Z<sub>c</sub>. Les fonctions F<sub>b</sub> et F<sub>c</sub> attachées à B et C, reçoivent le message WM\_PAINT, dont les paramètres décrivent Z<sub>b</sub> et Z<sub>c</sub> (Fig. V.6).

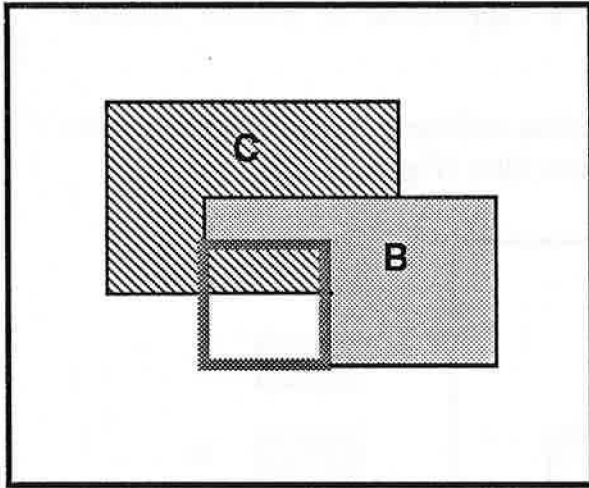


La fenêtre A est fermée, le système demande  
à B de repeindre Z<sub>b</sub>  
à C de repeindre Z<sub>c</sub>

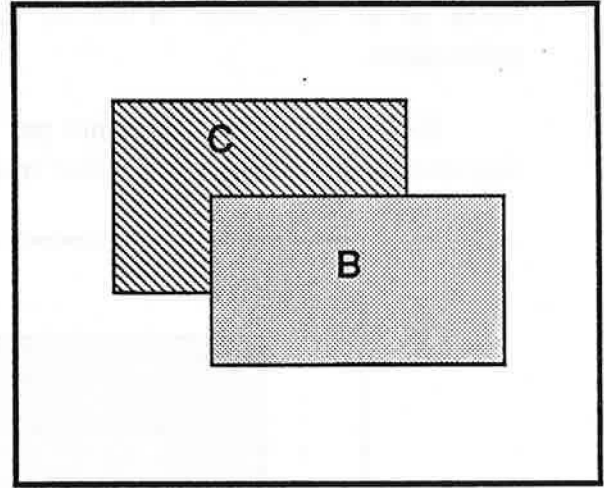
V : Réalisation

Fig V.6 : Zones de B et C à repeindre

Si  $F_c$  est la première à traiter le message, le scénario de mise à jour est le suivant (Fig. V.7).



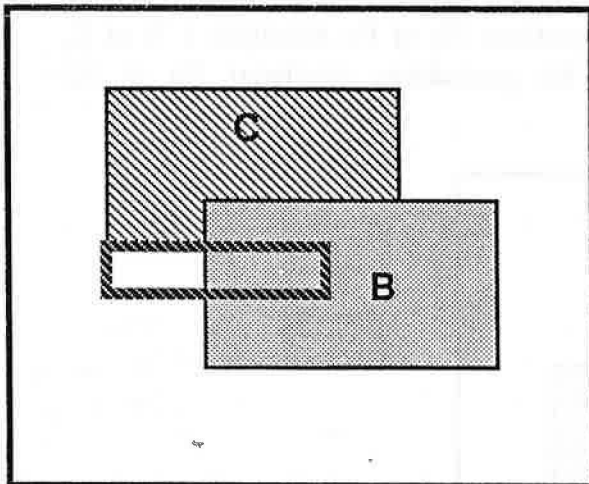
C régénère sa fenêtre en premier



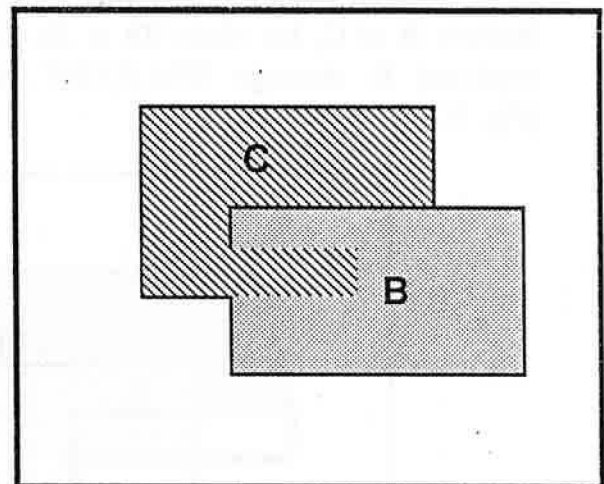
B régénère Zb

FIG V.7 : Scénario pour lequel C est repeinte en premier

Si c'est  $F_b$  qui reçoit le premier message, le résultat change (Fig. V.8).



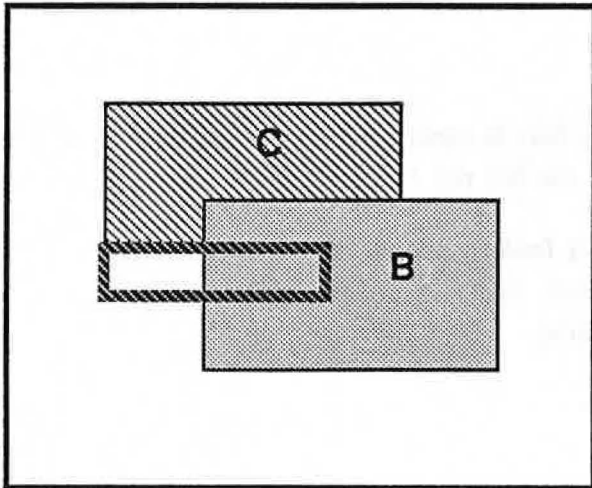
B régénère Zb



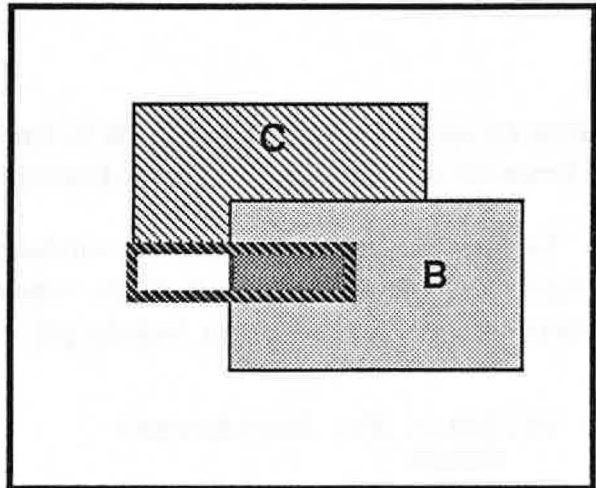
C régénère Zc

Fig V.8 : Scénario pour lequel C est repeinte en premier

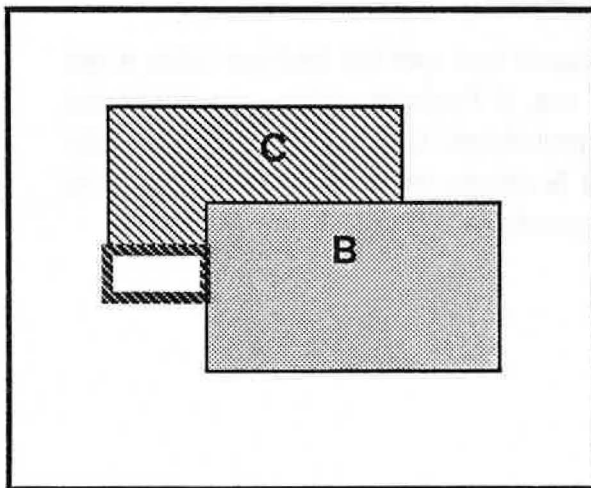
L'ordre des messages ne peut être ni prévu, ni forcé. Aussi, lorsqu'une fenêtre est mise à jour, elle doit valider la zone correspondante dans chacune des fenêtres qu'elle recouvre. Ainsi, le système masque la fenêtre fraîchement repeinte évitant ainsi, toute pollution inopinée. Si Fb traite le message en premier, le scénario est le suivant (Fig. V.9).



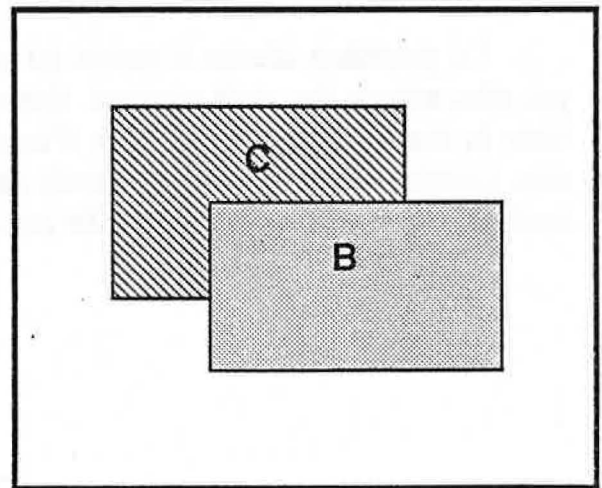
B régénère Zb



Zône validée par B



Zc n'empiète plus sur B



C ne pollue plus B

Fig V.9 : Restauration correcte

V : Réalisation

Sur réception de WM\_PAINT, pour la zone Rect, le traitement est alors

```

début
repeindre la fenêtre
FenToEcran(Rect, RectEcran)
pour toutes les fenêtres soeurs Fs faire Valider( Fs, RectEcran);
fin

```

La zone est exprimée dans le repère de la fenêtre. Il faut la ramener dans le repère de l'écran qui est commun à toutes les fenêtres. Ceci est fait par *FenToEcran()*.

La fonction *Valider()* permet de valider pour la fenêtre donnée en paramètre, la région de l'écran qui vient d'être repeinte. Les fonctions *GetRectEcran()*, *Intersection()*, *ValiderRect()* sont fournies par le système.

```

Valider( Fs, RectEcran)
  début
  GetRectEcran( Fs, RectFenEcran);
  Intersection( RectEcran, RectFenEcran, InterRectEcran);
  EcranToFen(InterRectEcran, InterRectFen);
  ValiderRect( Fs, InterRectFen);
  fin

```

Le programmeur n'est pas censé connaître les fenêtres visibles lorsqu'il rafraîchit une fenêtre. Pourtant, il doit valider une région dans chacune d'entre elles. Pour répondre à ce problème, le système offre la fonction *EnumChildWindows()* qui énumère toutes les sœurs d'une fenêtre.

La procédure décrite ci-dessus est suffisante tant que les fenêtres filles n'ont pas elles-mêmes des *child windows*. Dans ce cas, il faudrait valider récursivement toute la descendance. Ce qui pose d'autres problèmes. Comme ce n'est pas utile dans CRMM, et que cette tâche devrait être à la charge de tout bon gestionnaire de fenêtres, le présent rapport ne détaille pas les procédures à utiliser.

## 2 PRESENTATION GENERALE DES MODULES DE CRMM

Le logiciel est découpé en quatre modules principaux (Fig. V.10) :

- GBF qui gère la base des fiches,
- RGM qui s'occupe de la communication avec les autres BBGA,
- IU chargé de l'interface utilisateur,
- UA qui coordonne le tout, reçoit les requêtes de RGM et IU, répond à IU, fait appel aux modules GBF, RGM et IU en complétant les requêtes, supprimant les références locales et vérifiant les droits.

Le document de référence définit et spécifie chacun d'eux. Cette thèse s'intéresse particulièrement à l'interface utilisateur. Deux rapports à venir reprendront les autres modules de CRMM.

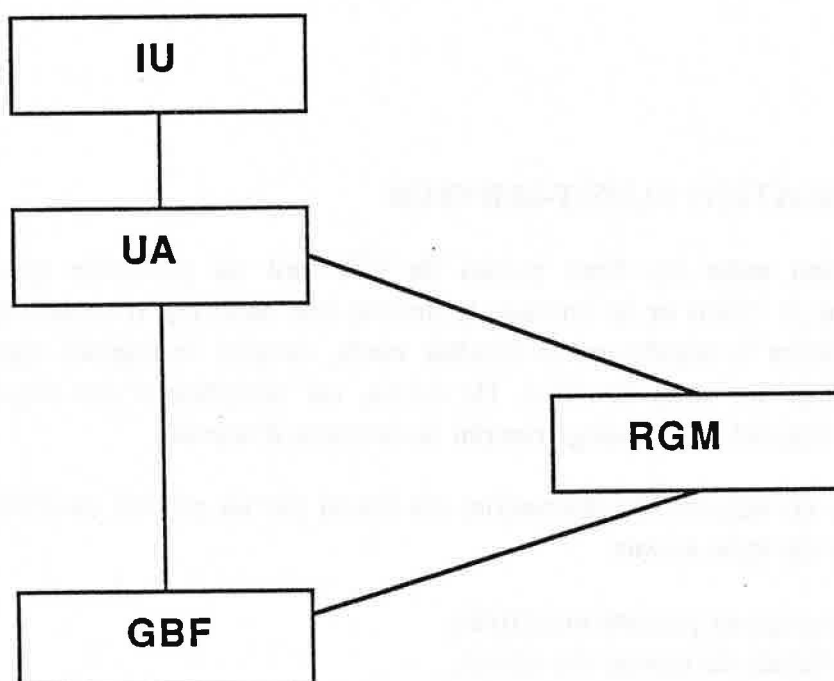


Fig V.10 : Les modules de CRMM

### 3 ECLATEMENT DU MODULE IU

Le module IU est en deux parties, une sur le serveur **IU serveur**, et une sur le poste de travail **IU client** (Fig. V.11).

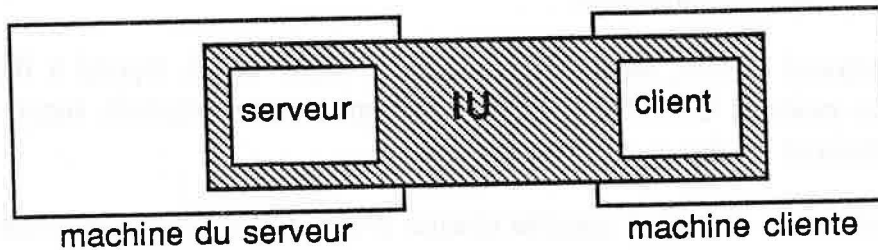


Fig V.11 : Eclatement du module IU

#### 3.1 COMMUNICATION CLIENT-SERVEUR

La communication entre les deux parties de l'IU suit un protocole spécifique nommé P31. Sur le client et le serveur, se trouve une structure d'accueil qui est remplie pour décrire la requête ou le résultat voulu, ensuite, le logiciel appelle la fonction de codage sur cette structure. De même, sur réception d'une requête ou d'un résultat, le logiciel de décodage remplit la structure d'accueil.

Le service de transfert d'informations est fourni par un produit du CNET qui suit un protocole de style *kermit*:

- découpage en paquets numérotés,
- possibilités de reprise sur erreur,
- fonctionnement sur différents supports : X25 ou asynchrone v24.

### 3.2 LES OBJETS IU SERVEUR

L'IU serveur gère une copie de la liste de sélection, il s'agit de la liste des numéros locaux obtenue par la précédente opération de sélection.

### 3.3 LES OBJETS IU CLIENT

**Liste des fiches sélectionnées :** elle est modifiée à chaque opération de sélection. Elle contient les numéros locaux des fiches concernées, leur identifiant global et le nom générique des fichiers qui les décrivent.

**Liste des fiches disponibles :** cela donne les fiches disponibles sur la machine du client. Il s'agit d'une liste circulaire. Toutes les fiches qui sont reçues sur la machine cliente sont inscrites dans cette liste. L'insertion d'une fiche est gérée de la façon suivante :

```

i ← indice courant d'insertion
N ← nombre d'éléments de la liste
Tant que fiche i affichée ou sélectionnée
faire
    début
        i ← (i + 1) modulo N
    fin
insérer la fiche

```

**Liste des fiches précédentes :** cette liste mémorise les dernières fiches lues par l'utilisateur. Une fiche est considérée lue si elle a été affichée dans une fenêtre et si cette dernière a été fermée. Il s'agit d'une liste circulaire qui est référencée par un pointeur d'insertion. Elle donne le numéro local des fiches.

**Contenu de la boîte :** il s'agit d'un tableau qui contient pour chaque message, le nom externe du message et le nom interne du message.

**Liste des marques :** ce tableau qui donne le numéro local de la fiche et la valeur de la marque correspondante.

**Liste des conférences :** cette table contient le nom des conférences et le nombre des fiches non lues.

**Liste des fiches lues :** elle contient le numéro le plus élevé correspondant à



une fiche lue, la liste des numéros de fiches lues supérieurs à la valeur précédente, et le numéros de fiche le plus élevé de la conférence.

**Droits des utilisateurs :** il s'agit de booléens, un pour le droit à écrire, un pour le droit d'éditeur et un pour le rôle d'administrateur.

**Fiches disponibles :** la machine du client stocke sur disque les fiches actuellement présentes à l'écran et d'autres susceptible d'être demandées. Chacune d'elles est décrite par trois fichiers de même nom, placés dans des répertoires différents :

fichier en-tête il contient le nombre de lignes de texte, le nombre de caractères maximum d'une ligne, le nom de l'auteur, le nom de l'éditeur, la date de prise en compte, l'objet, le nombre et la liste des liens ascendants, des liens descendants et des mots clés.

fichier index il repère le début de chaque ligne de texte. Ceci est utile lors de mouvement

fichier contenu il s'agit du corps de la fiche.

La procédure d'affichage d'une fiche est la suivante :

```

prem_lig ← première ligne visible
nbre_lig ← nombre de lignes visibles
prem_car ← premier caractère visible
nbre_car ← nombre de caractères visibles

si en-tête visible alors
  début
  lire en-tête dans fichier en-tête
  écrire en-tête
  fin

pour l ← prem_lig à prem_lig + nbre_lig
  début
  pos ← index l dans fichier index
  lire ligne dans fichier contenu à partir de pos
  pour c ← prem_car à prem_car + nbre_car
    début
    afficher car c de ligne
    fin
  fin

```

### 3.4 LES FENETRES

Les fonctionnalités du service de conférence sont offertes dans différentes fenêtres dont la liste suit. Ce sont soit des *child windows* entièrement contrôlées par l'application, soit des *Dialog Box* dont la gestion est en partie prise en charge par le système Windows. Les traitements d'une *child window* et d'une *dialog box* sont donnés en exemple, pour les autres, seules sont données leurs différences.

*La fenêtre de lecture* est une *child window*, elle est pourvue de barre de déplacement, d'un ascenseur horizontal, d'un ascenseur vertical, d'une boîte de taille (*size box*). Optionnellement, l'en-tête peut être visible. La taille et le remplissage de la fenêtre sont contrôlés par le logiciel d'application. La fonction qui gère la fenêtre traite les messages suivants :

**INIT\_CONT** Il s'agit d'un message propre à l'application. Il est généré lorsque l'utilisateur demande la lecture de la fiche. La fonction lit le fichier index de la fiche concernée et initialise les variables décrivant la fiche (nombre de lignes, en-tête....)

**WM\_SIZE** La fenêtre reçoit ce message chaque fois que la taille de la fenêtre change. Les paramètres donnent les nouvelles dimensions. Il faut changer les nombres de lignes et de caractères visibles, vérifier que la taille n'est pas trop grande et optimiser son remplissage (cela influe sur la première ligne visible et le premier caractère visible).

**WM\_HSCROLL** Il est généré lorsque l'usager touche à l'ascenseur horizontal. Les paramètres précisent la nature de l'action (étendue et sens du déplacement). Cela modifie le premier caractère visible.

**WM\_VSCROLL** Il concerne l'ascenseur vertical. Les paramètres donnent l'étendue et le sens du déplacement. Cela influe sur la première ligne visible.

**WM\_SYSCOMMAND** Si le paramètre est **SC\_CLOSE**, il signale la demande de fermeture de la fenêtre. Il faut alors procéder aux opérations correspondantes.

**WM\_PAINT** Il est reçu lorsqu'une zone de la fenêtre n'est plus à jour et doit être repeinte.

la *fenêtre de sélection de fiches* est une *child window*, elle gère les messages de la fenêtre précédente. En plus, elle saisit les caractères entrés au clavier pour former les critères de sélection. La saisie et l'affichage sont contrôlés caractère par caractère. Ceci complique le traitement, choisir une *Dialog Box* aurait facilité le programme, mais contrôler la saisie permet de prévenir immédiatement l'utilisateur de ses erreurs.

Un critère est composé d'atomes, par exemple, le critère :

Auteur = BRUN + PAYS

contient les atomes BRUN et PAYS. Chaque fin d'atome dans un critère provoque le contrôle de sa validité. Ceci est fait sur réception des événements :

**VK\_UP** l'utilisateur appuie sur la touche flèche vers le haut pour passer au critère de la ligne précédente.

**VK\_DOWN** l'utilisateur appuie sur la touche flèche vers le bas pour passer au critère de la ligne suivante.

**VK\_RIGHT** l'utilisateur appuie sur la touche flèche vers la droite pour passer au caractère suivant, ce qui peut provoquer la sortie de l'atome courant.

**VK\_LEFT** l'utilisateur appuie sur la touche flèche vers la gauche pour passer au caractère suivant, ce qui peut provoquer la sortie de l'atome courant.

**VK\_RETURN** l'utilisateur appuie sur la touche retour-chariot pour passer au premier caractère de la ligne suivante.

**VK\_BACK** l'utilisateur efface le caractère courant ce qui peut fusionner les deux atomes si le caractère effacé est OU.

**OU** frappe d'un caractère OU qui termine un atome et entame un suivant. Pour l'instant, OU = '+'

**WM\_LBUTTONDOWN** déplacement dans la fenêtre de saisie grâce à la souris qui pointe la position d'arrivée.

L'application vérifie selon la nature du critère, s'il s'agit bien :

- d'une date,
- d'un nom local,
- d'un nom complet,
- d'un identifiant global
- d'un numéro local,
- d'un lien.

la *fenêtre de sélection de liens*, la fonction qui gère cette *child window* est très proche de celle de la fenêtre précédente. Seuls changent le nombre et la valeur des critères de sélection.

la *fenêtre boîte aux lettres*, cette fenêtre est une *Dialog Box* qui comprend les contrôles suivants :

- une *List Box* qui donne les messages contenus dans la boîte aux lettres.
- un bouton qui contient la chaîne **Répondre**,
- un bouton pour **Lire un message**
- un bouton pour **Annuler** la commande

la *fenêtre marque*, c'est une *Dialog Box* comme les fenêtres suivantes :

la *fenêtre état des conférences*

la *fenêtre de résultat d'opération*

la *fenêtre de copie de fiche*

la *fenêtre de création de fiche*

### 3.5 LES MENUS

CRMM utilise deux menus, le premier (**Menu principal**) est tout le temps présent au sommet de la fenêtre de l'application, sauf lors de sélections de liens ou de fiches. Pendant ces phases, il laisse la place au **Menu de sélection**.

Le menu de sélection est composé des choix suivants :

**CRMM** Il s'agit d'un menu déroulant, lorsqu'il est choisi il propose :

- *Etat des conférences*, pour avoir le nom des conférences auxquelles l'utilisateur est abonné et le nombre de fiches non lues pour chacune d'elles.
- *Connexion* permet de se connecter à une conférence.
- *Quitter* provoque la fin de la session.

**Fiche** ce menu déroulant donne accès aux opérations suivantes

- *lire*
- *effacer*

- *détruire*
- *copier*

**Options** ce menu déroulant offre les possibilités suivantes :

- *En-tête* des fiches, il est visible ou non.

**Marques** Ce choix appelle la fenêtre des marques.

**Imprimatur** Il provoque l'affichage de la fenêtre de la boîte aux lettres (pour les auteurs ou les éditeurs).

**Suivante** Ce menu déroulant demande la fiche suivante

- *dans la sélection* courante, ou dans la liste des
- *non lues*

Le menu de sélection apparaît en même temps que les fenêtres de sélection de liens ou de fiches, donc après une demande d'opération pour que l'utilisateur précise les fiches ou liens concernés. Il propose :

**QUITTER** pour clore la saisie des critères de sélection.

- *Lancer opération*, la saisie se termine par le lancement de l'opération sur les fiches ou liens sélectionnés,
- *Annuler* : retour au menu principal sans effectuer l'opération.

**INITIALISATION** ce menu déroulant permet d'agir sur les critères de sélection saisis :

- *Remise à zéro* les efface tous.
- *Sélection précédente* affiche les critères de la dernière opération.

**NOUVEAUTE** (seulement pour la sélection de fiches) l'utilisateur ajoute éventuellement un critère

- *fiches non lues*, les fiches sélectionnées devront être nouvelles,
- *fiches déjà lues*, les fiches sont anciennes

- *indifférent* les fiches ont été lues ou non.



**CRMM++** 141

---

**NOUVELLES FONCTIONNALITES** 142

**IMPLANTATION** 145

**NOUVEAUX SERVICES** 145

**GENERATEUR DE CONFERENCE** 148





## Chapitre 6

### CRMM++

Le projet CRMM date de deux ans seulement. Il couvre plusieurs domaines de l'informatique :

- messagerie X400,
- communication en mode connecté,
- communication de groupe,
- bases de données,
- interface utilisateur.

De ce fait, la première version du modèle et du produit ne prétend pas être définitive. D'abord, toutes les possibilités actuelles des domaines évoqués au-dessus n'ont pas été exploitées. Ensuite et surtout, ces domaines sont en pleine évolution.

Aussi, les futures versions de CRMM devront présenter certaines améliorations<sup>1</sup>. Certaines concernent la version actuelle du produit, elle ne sont que de simples retouches sans remises en cause profondes de l'architecture et du modèle CRMM, d'autres sont plus fondamentales, elles impliquent une refonte importante. En fait, elles s'appuient sur des notions qui sont aujourd'hui à l'étude, et sont étroitement liées aux travaux en cours. Elles déboucheront sur une nouvelle version de CRMM.

---

1 L'expression ++CRMM semblerait idéale (les améliorations sont faites avant que le produit ne soit délivré), mais la plupart des progrès dépendent de travaux en cours dans les divers organismes de normalisation ou institutions de recherche.

Ce chapitre présente des suggestions d'évolutions souhaitables et prévisibles pour notre système de conférences. Elles concernent soit des fonctionnalités nouvelles offertes à l'utilisateur, soit l'utilisation de nouveaux matériels et logiciels de base, soit une prise en compte de nouveaux services de bureautique.

La première partie de cette section décrit de nouveaux services qu'il est souhaitable d'offrir, tant au niveau fonctionnel du système de conférences que dans l'interface qui en est proposée à l'utilisateur. La suite du chapitre montre les progrès envisageables dans la réalisation, en tirant profit de nouveaux matériels et de nouveaux services logiciels. Ensuite, une architecture différente est présentée, elle tient compte des nouveaux services accessibles par l'utilisateur. Enfin, la dernière partie présente une extension importante du système de conférences qui facilite la génération de conférences.

## 1 NOUVELLES FONCTIONNALITES

CRMM tel qu'il est décrit dans ce document, est fortement influencé par l'expérience et les souhaits d'informaticiens utilisateurs de systèmes de conférences actuels. Toutefois, il est vraisemblable qu'un public plus large ne ressentira pas les mêmes besoins. C'est pourquoi des choix *a priori* intéressants peuvent être remis en cause par les utilisateurs. Ainsi, les relations et la structure de la base des messages qui en résultent, seront enrichis (nouvelles classes de liens, opérations supplémentaires) ou au contraire, simplifiés. La phase de sélection et notamment la saisie des requêtes peuvent évoluer dans les mêmes directions.

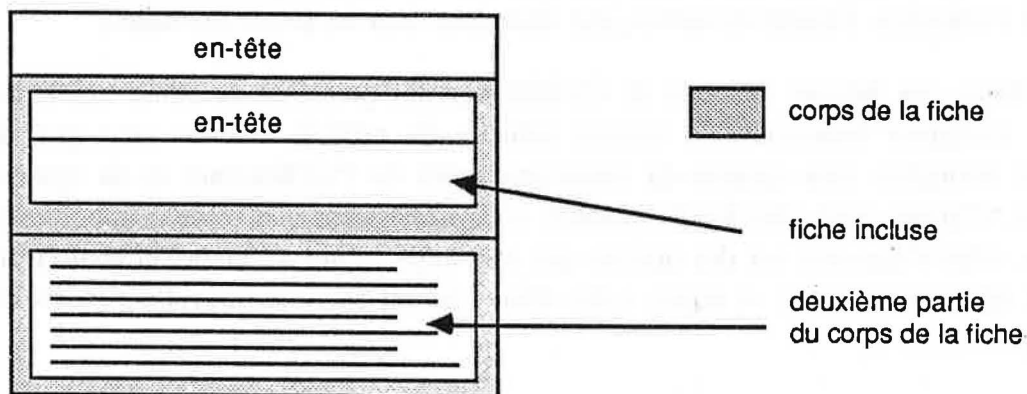


Fig VI.1 : Fiche au corps structuré

Parmi les fonctionnalités nouvelles qui sont souhaitables, on peut signaler l'enrichissement des messages échangés entre utilisateurs. D'abord, le contenu de la fiche peut être structuré en plusieurs corps, eux-mêmes constitués de fiches (Fig. VI.1). Ensuite, les messages de CRMM sont aujourd'hui purement textuels, il serait bien qu'ils puissent supporter différents codages comme la voix ou les images. La compatibilité avec les messages du courrier interpersonnel serait un atout supplémentaire. Cela faciliterait les échanges entre les deux services.

L'interface peut être améliorée sans remettre en cause les services offerts par le système de conférence, plusieurs options pourraient être proposées. Par exemple, l'intégration de CRMM avec d'autres utilitaires à la disposition de l'utilisateur comme des traitements de texte, des tableurs ou des outils graphiques, doit être favorisée. Ainsi, les usagers profiteraient des qualités intrinsèques de chaque outil. Une bonne coopération entre les différents produits ne dépend pas entièrement des concepteurs de CRMM, mais aussi de l'apparition de normes pour les formats de données échangées entre applications d'un même intégrateur.

Actuellement, la structure de la base des fiches n'apparaît à l'utilisateur qu'à travers les en-têtes des messages (Fig. VI.2) qui donnent toutes ses relations avec d'autres fiches de la base. Une représentation graphique serait plus expressive (Fig. VI.3) et elle ferait apparaître la structure dans un voisinage plus large. Cette visualisation d'une structure de documents est à rapprocher de la notion d'*Hypertext* [CONKLIN87]. Les travaux en cours dans ce domaine débouchent sur des idées et des produits intéressants dans le domaine de l'interface utilisateur comme l'*Hypercard* sur Macintosh [GOODMAN87].

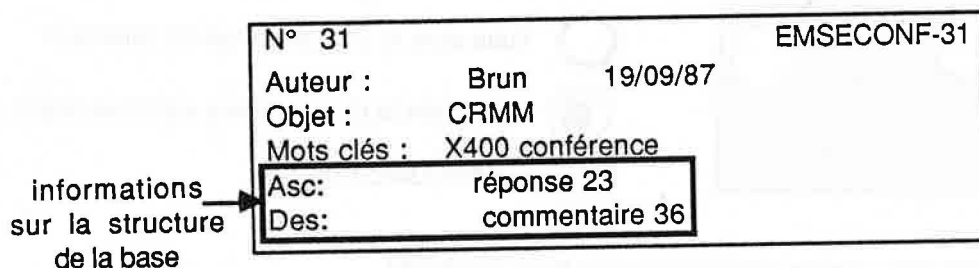


Fig VI.2 : Liens dans un en-tête

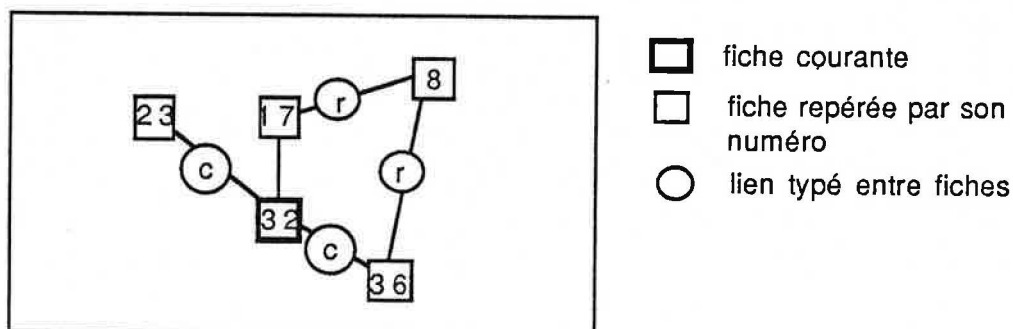


Fig VI.3 : Proposition de représentation graphique de la structure

Il serait bien que chaque utilisateur ait une vision personnelle de la base. Aujourd'hui, ceci se limite aux fiches non lues et aux marques, mais il est possible de l'étendre. Par exemple, un usager peut souhaiter "couper" une partie de la structure (une conversation) pour ne plus être informé de l'arrivée de messages s'y rapportant (Fig. VI.4). Réciproquement, une conversation peut intéresser un utilisateur qui souhaite alors être prévenu lorsque de nouveaux messages viennent s'y rajouter. Il peut par exemple guetter l'arrivée de réponses à une question posée par une fiche donnée.

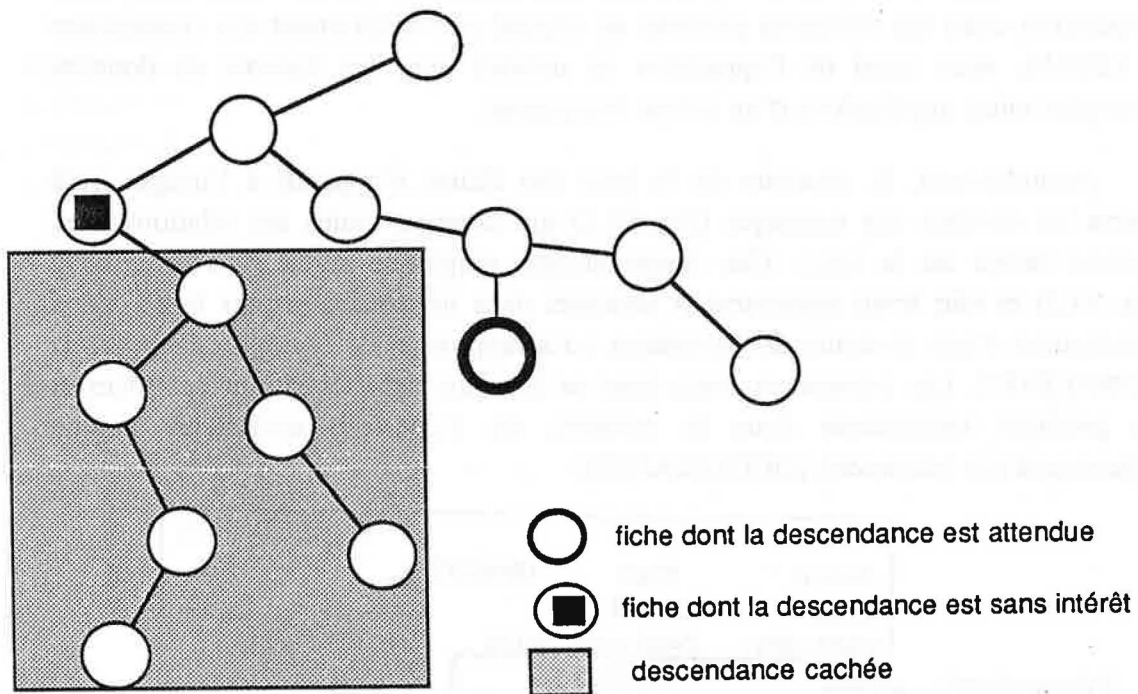


Fig VI.4 : Vision personnelle de la base des fiches

## 2 IMPLANTATION

La réalisation profitera des progrès du matériel et des logiciels de services. Comme tous les autres produits informatiques, CRMM bénéficiera de composants plus rapides, plus intégrés et moins chers. Les programmes écrits à l'aide de bibliothèques du *toolkit* supporteront en partie une évolution du matériel. Plus spécifiquement, le système de conférences est intéressé par un profond changement du système d'exploitations des postes de travail sur ordinateurs personnels compatibles. Certes, cela imposera de récrire une partie importante des programmes mais en contrepartie, ces nouveaux systèmes offriront des facilités pour gérer les communication entre la station de travail et le serveur, notamment, ils supporteront le multi-traitement. Un protocole plus puissant deviendra alors possible. Il permettra des anticipations plus élaborées des besoins de l'utilisateur ce qui réduira son attente.

## 3 NOUVEAUX SERVICES

La session de travail du CCITT en cours et celle qui débute en 1988 définiront de nouveaux services comme les listes de distribution, l'annuaire et l'archivage. Ces services doivent être pris en compte par CRMM. Ils seront accessibles par protocoles normalisés (Fig. VI.5). Par exemple, l'archivage des fiches de la conférence pourra être pris en charge par un serveur dédié. Toutes les opérations sur la base se feront par requêtes qui suivent le protocole d'accès au service. Actuellement, les études sur ce serveur sont orientées vers les messages au format du courrier interpersonnel, mais les recommandations prendront aussi en compte des attributs des futurs services de messagerie comme les conférences.

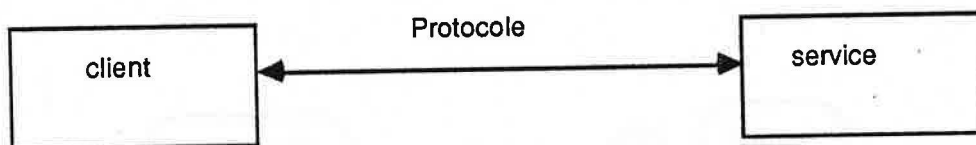


Fig VI.5 : Protocole d'accès à un service

Le service rendu par l'annuaire est le suivant. Le client donne un nom, il répond par les attributs qui caractérisent l'entité nommée, par exemple, son adresse. La signification de l'attribut et l'utilisation qui en est faite ne concernent pas l'annuaire. Dans le cas de CRMM, l'annuaire pourrait servir à contrôler les droits des usagers et ainsi à réaliser une nouvelle fonctionnalité, correspondante au visa d'expédition du courrier de personne à personne. Ceci prend en compte l'organisation et le règlement intérieur d'une entreprise utilisatrice de la messagerie. Il s'agit de contrôler la diffusion de l'information hors de l'entreprise, en imposant un visa de sortie par une personne autorisée. Le scénario d'une requête d'insertion d'une fiche devient (Fig. VI.6) :

1. requête d'insertion de l'auteur vers le service de conférences,
2. requête du service de conférences vers l'annuaire en donnant le nom de l'auteur,
3. réponse de l'annuaire qui provoque l'un des traitements suivants
  - si la requête de l'auteur doit être visée
    4. envoi vers l'entité qui accorde les visas
    5. en cas d'accord, le message est pris en compte par le système de conférences, il sera ajouté à la base (l'annuaire intervenant de la même façon pour décider d'une éventuelle soumission à un éditeur).
  - si la demande de l'auteur ne nécessite pas de visa, le service de conférence prend directement en compte la requête.

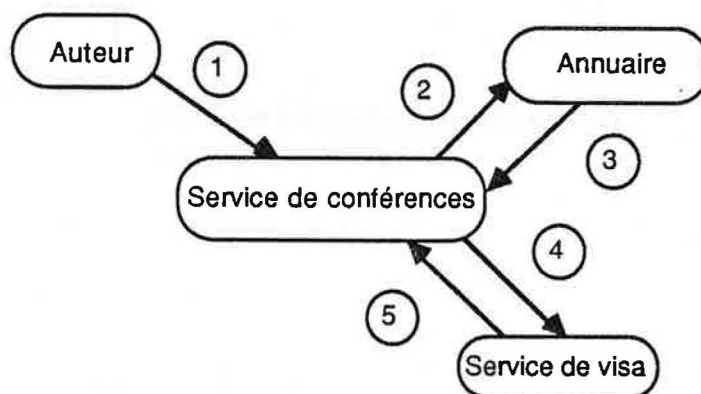


Fig VI.6 : Requête d'insertion de fiche avec service d'annuaire

Un service de listes de distribution serait utile pour gérer la diffusion des messages à l'ensemble des sites de la conférences. Il prendrait en compte les modifications de la liste (insertion, suppression d'un site) et la diffusion de fiches. Dans le cas de CRMM, les services demandés au service de liste de distribution ne seraient pas très complexes. En effet, la liste des sites est simple, elle n'inclut pas de sous-liste qui provoqueraient des boucles dans la distribution, de plus CRMM ne prévoit pas de faire suivre des messages, opération délicate dans un contexte de diffusion.

Outre les services utilisables par CRMM, le CCITT va recommander d'autres services directement accessibles par l'utilisateur. Cette multiplication des services se traduit par autant d'Agent Utilisateurs (UA). Ces accès différents rendent difficile une bonne interface avec l'usager. Il serait bien de donner un point d'entrée unique pour tous ces services de bureautique pour qu'ils soient utilisables conjointement. L'usager ne serait représenté que par un seul UA multifonction. (Fig. VI.7)

Ce modèle se conjugue bien avec l'idée de l'UA réparti, de même l'architecture client-serveur est bien adaptée à la coopération des services (Fig. VI.8).

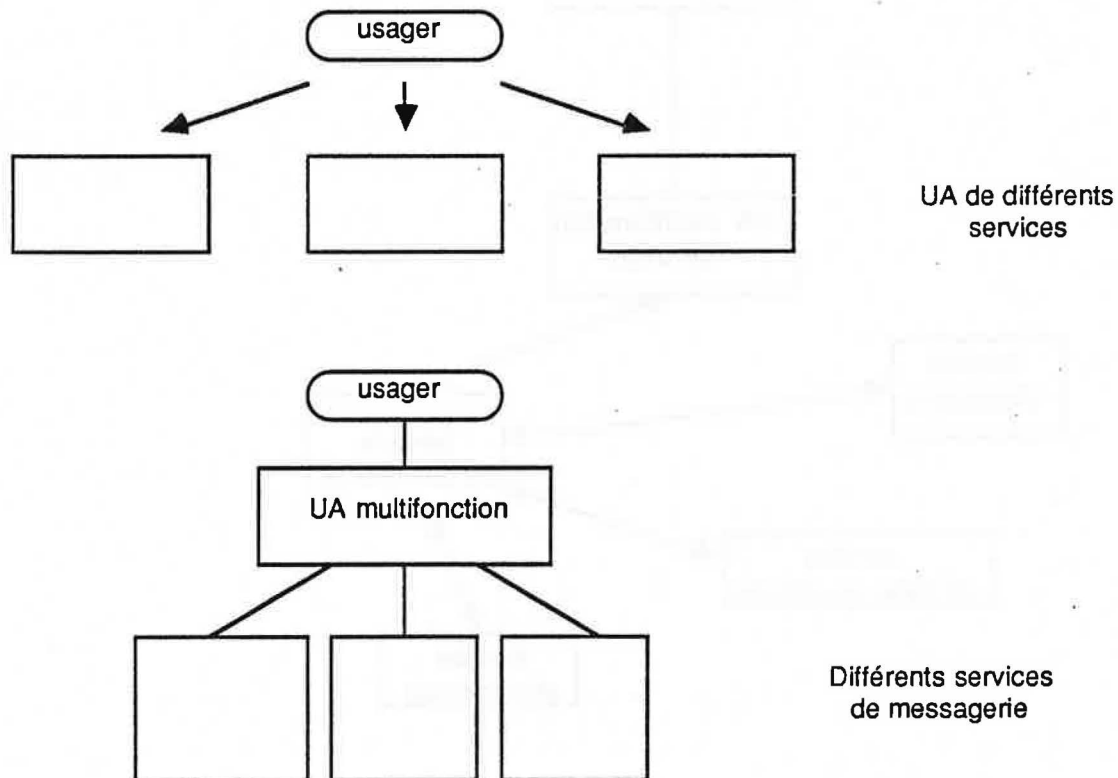


Fig VI.7 : UA multifonction



## 4 GENERATEUR DE CONFERENCE

Tout ce qui précède laisse supposer, à juste titre d'ailleurs, que la mise en œuvre d'une conférence n'est pas une tâche aisée. En effet, il faut donner une liste de sites avec les droits qu'ils peuvent accorder, définir les liens qui seront disponibles, mettre en place une comptabilité, établir une facturation (qui paye le service ?), installer la conférence sur chaque site.... De même, un site qui souhaite s'intégrer dans une conférence en cours n'a pas la partie belle. Pour cela, il serait bien de définir un générateur de conférence. Chaque site appartenant au système de conférence, pourrait s'intégrer à une conférence existante et à partir de chaque site, un usager aurait la possibilité de créer une conférence.

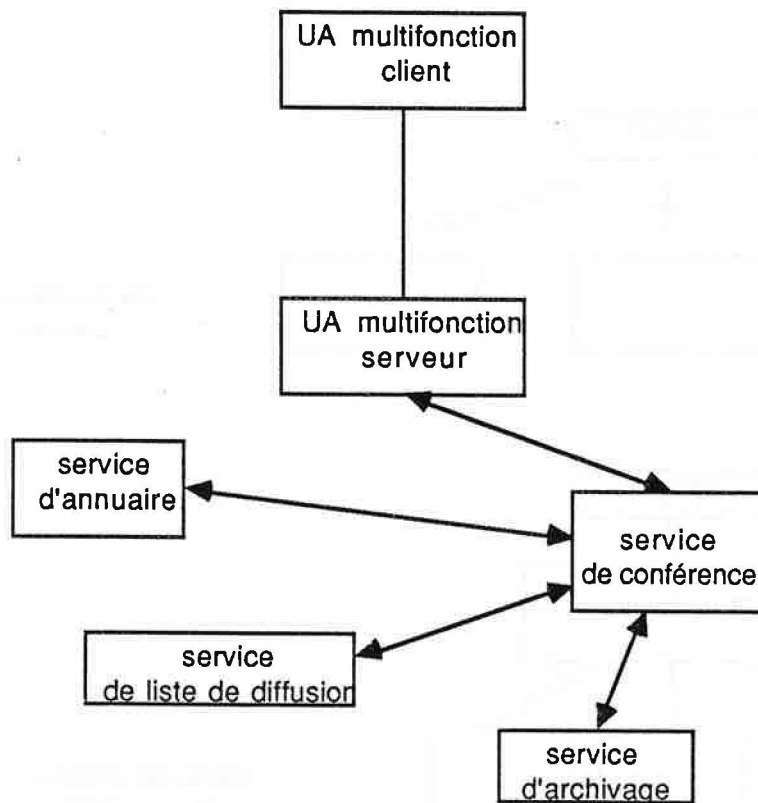


Fig VI.8 : UA éclaté et coopération de services





## Chapitre 7

### CONCLUSION

Les conférences informatiques sont promises à un bel avenir, le développement du courrier électronique va révéler le besoin d'un outil de communication de groupe. De plus, elles permettent de couvrir un large champ d'applications.

Les idées présentées dans cette thèse serviront certainement à cette extension. La répartition est nécessaire pour réduire les coûts de communication, le respect des normes en vigueur (OSI) permet des échanges entre systèmes hétérogènes, la messagerie X400 (boîtes aux lettres et mode non connecté) permet une utilisation rationnelle et efficace des supports de communications, le contrôle des droits des utilisateurs est impératif à une bonne vie du groupe des membres des conférences et la prise en compte des relations entre messages permet des échanges plus riches en informations. Enfin, il est nécessaire de faciliter l'accès au produit pour les utilisateurs. Ceci passe par l'étude de l'interface du produit et par une bonne intégration du service de conférences parmi les services de messagerie et les autres outils dont dispose l'utilisateur. Cette effort d'intégration se manifeste dans l'élaboration du modèle (UA multifonction), dans le choix des outils (intégrateurs) et dans la réalisation (cohérence dans la présentation, échanges de données entre applications).

Outre les idées, le projet CRMM a donné naissance à un produit qui mérite un succès d'utilisation par les services qu'il offre et par l'interface qu'il en propose.

CRMM va être soumis à normalisation dans le cadre des travaux du CCITT en tant que nouveau service de messagerie sous le label TCAO pour **TéléConférence Assistée par Ordinateur**. Il est certain que les propositions faites dans le projet CRMM ne seront pas retenues en totalité pour les recommandations finales. Toutefois, les idées et l'expérience de CRMM contribueront certainement à la version qui résultera de ces travaux.

Pour ce qui me concerne, l'intérêt de cette thèse tient d'abord aux sujets qu'elle aborde (messagerie X400, base de données, répartition et interface utilisateur). De plus, la partie réalisation qu'inclut la thèse est intéressante, elle met en jeu des outils nouveaux comme les intégrateurs. Mais le principal intérêt de cette thèse réside dans la formation à la recherche que j'ai reçue. Pour terminer, je voudrais particulièrement insister sur l'excellent encadrement dont j'ai bénéficié à l'Ecole des Mines. J'espère avoir tiré profit de sa compétence et de sa disponibilité.

## BIBLIOGRAPHIE

- [BENFORD87] Benford S., Onions J.  
Pilot Distribution Lists - agents and directories  
Proceedings of the IFIP 6.5 International Working Conference on MHS  
Munich, Avril 1987
- [BOHN87] L. Bohn, D. Weinberger  
Why not have it all ?  
juillet 1987  
Unix Review vol 5 Number 7 pp 28-34
- [BRUN86] Brun Ph., Pays P.A. et al.  
CRMM : Conférence Répartie en Mode Messagerie  
Acte du congrès DNAC 1987  
Paris, Novembre 1986
- [BRUN86] Brun Ph., Pays P.A. et al.  
CRMM: Document de référence  
ENSMSE : Rapport de recherche 86-08, Septembre 1986
- [BRUN87] Brun Ph.  
Interface utilisateur pour produits de messagerie  
Acte du congrès DNAC 1987  
Paris, Septembre 1987
- [CAROLL83] Caroll J.M.  
Presentation and Form in User-Interface Architecture  
Décembre 1983, BYTE pp 113-122

- [CCITT84] Série de recommandations X400
- [CHESS87] Chess D.M., Cowlshaw M.F.  
A Large-scale Computer Conferencing System  
IBM Systems Journal, Janvier 1987, Vol 26 Num 1 pp 138-153
- [COMER86] D.C. Comer, L.L. Peterson  
Conversation-Based Mail  
ACM Transactions on Computer Systems,  
Novembre 86, Vol 4, Num 4, pp 299-319
- [CONKLIN86] Conklin J.  
A Survey of Hypertext  
Septembre 1987, Computer, Vol 20 num 9 pp 17-41
- [COUTAZ85] Coutaz J.  
A Layout Abstraction for Interactive Interfaces  
SIGCHI Bulletin, Janvier 1985, pp 18-24
- [COUTAZ86] Coutaz J.  
Abstractions pour Interfaces Interactives  
TSI, Avril 1986, pp 239-250
- [DANIELSEN86] Danielsen T., Pankoke U., Pays P.A. et al.  
The Amigo Project: Advanced Group Communication Model for  
Computer-Based Communication Environment  
acte du congrès CSCW 86  
Austin, Décembre 1986
- [DENNING82] Denning P.  
Electronic Junk  
Communications of the ACM, 1982 Vol 25, Num 3
- [DJEBALI88] Djebali A.  
Rapport de Thèse (à paraître)  
Université de Paris VI  
Courant 1988
- [EDWARDS83] Edward S.  
Why is Software So Hard to Use ?  
Décembre 1983, Byte pp 127-138  
.BP
- [FELDMAN86] Feldman M.S.  
Constraints on Communication and Electronic Mail

acte du congrès CSCW 86  
Austin, Décembre 1986

- [GOODMAN] Goodman D.  
The Two Faces of Hypercard  
Octobre 1987, MacWorld pp 122-129
- [GUEDJ79] Edité par Guedj R.A., et al.  
Methodology of Interaction  
IFIP Workshop on Methodology of Interaction  
IFIP W. G. 5.2 CAD Mai 1979, Seillac
- [HERBACH83] Herbach M.  
How Is a Computer Like an Elephant  
Décembre 1983, Byte pp 247-258
- [HILTZ85] S.R. Hiltz, Turoff M.  
Structuring Computer-Mediated Communication Systems to Avoid  
Information Overload  
Communication de l'ACM, Juillet 85, Vol 28 Num 7 pp 680-689
- [HODGSON85] Hodgson G.M., Ruth S.R.  
Use of Menus in Design of On-line Systems: a Retrospective View  
SIGCHI Bulletin, Juillet 1985, Vol 17 Num 1 pp 16-22
- [HOPPER85] Edité par Hopper K., Newman I.A.  
Foundation For Human-Computer Communication  
Working Conference on The Future of Command Languages  
IFIP WG 2.6 Rome, Septembre 1985
- [HOPGOOD85] Edité par Hopgood F.R.A., Duce D.A., et al.  
Methodology of Window Management  
Proceedings of an Alvey Workshop at Cosener's House  
Abingdon UK, Avril 1985
- [HOUEY85] Houery N.  
Le projet Smartix  
Acte du congrès DNAC 1985  
Paris, Novembre 1985
- [HUTTEMA85] Huitema C.  
The Cosac Conferencing Experiment  
Proceedings of IFIP 6.5 International Working Conference on MHS  
Washington 1985



- [KAY79] Kay A.  
Smalltalk  
[GUEDJ80] pp 7-12
- [KERR82] Kerr E.B., Hiltz S.R.  
Computer-mediated Communication Systems - Status and Evolution  
Academic Press New York 1982
- [KINTZIG84] Kintzig C.  
Maquette du projet COSAC  
Janvier 84 Minis et Micros 202, pp 45-46
- [LYNCH86] Lynch G., Meads J.  
In Search of a User Interface Model: report on the SIGCHI workshop on  
user interface models  
SIGCHI Bulletin, Vol 18 Num 2 pp 25-33
- [MALONE85] Malone T.W.  
Designing Organizational Interfaces  
ACM SIGCHI Proceedings Avril 1985 pp 66-71
- [MEEK86] Meeks B.N.  
Conferencing Overview  
December 1985, Byte pp 169-184
- [MICROSOFT87]  
Windows : Manuel de référence du programmeur
- [PALME85] Palme J.  
Storage and Retrieval and Group Communication  
Draft paper ISO/TC 97/SC 18/WG 4
- [PALME86] Palme J.  
Conferencing Standards  
December 1985, Byte pp 187-194
- [PALME86] Palme J.  
Database Structure in PortaCOM  
December 1985, Byte pp 195-202
- [PALME87] Palme J.  
AMIGO: Document Storage and Retrieval  
Proceedings of IFIP 6.5 International Working Conference on MHS  
Munich Avril 1987

- [PAYS87] Pays P.A., Brun Ph.  
CRMM: A distributed Bulletin Board  
Proceedings of IFIP 6.5 International Working Conference on MHS  
Munich, Avril 1987
- [PIKE83] Pike R.  
Graphics in Overlapping Bitmap Layer  
Juillet 1983, Computer Graphics, Vol 17, Num 3
- [PRINZ87] Prinz W., Speth R.  
Group Communication and Related Aspects in Office Automation  
Proceedings of IFIP 6.5 International Working Conference on MHS  
Munich Avril 1987
- [SARAS87] Saras J.A., Huitema C.  
The Use of Distribution Lists in MHS  
Proceedings of IFIP 6.5 International Working Conference on MHS  
Munich Avril 1987
- [SHAPIRO85] Shapiro R.H., Anders  
Towards an ethics and etiquette for electronics mail  
éditeur : Rand Corp, 1700 Main Street  
p.o. Box 2138 Santa Monica  
ca 90406-2138, 1985
- [STEFIK87] Foster M., Foster G. et al.  
Beyond the chalkboard: Computers Support for Collaboration and  
Problem Solving in Meetings  
Communications of the ACM, January 1987 Vol 30 Num 1 pp 32-47
- [TAYLOR87] Taylor R.G., Mohammad J., Rubin T, Roberts T.  
Cosmos User Interface Team End Of Year Report
- [TALSKY] Talsky G.R.  
User Interface Offer Greater Power  
Décembre 1986, Mini-Micro Systems pp 81-95
- [WARFIELD83] Warfield R.W.  
The New Interface Technology. An Introduction to Winddows and Mice  
Décembre 1983, Byte pp 218-233

[WHITESIDE85] Whiteside J., Jones S., Levy P.S., Wixon D.  
User Performance with Command, Menu, and Iconic Interfaces  
ACM SIGCHI Proceedings, Avril 1985 pp 185-191

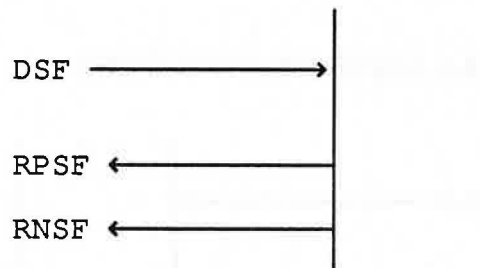
[ZENG88] Zeng Y.  
Rapport de Thèse (à paraître)  
Université de Paris VI  
Courant 1988

## ANNEXE

### LES PRIMITIVES DE CRMM

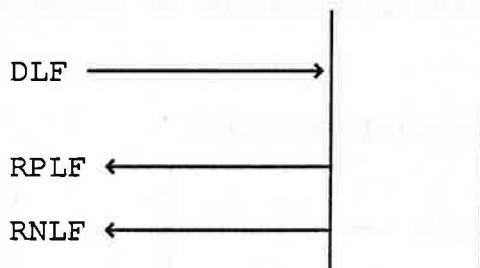
#### 1 PRIMITIVES DE BBGA OFFERTES AU BBUA

##### 1.1 SELECTION DES FICHES.



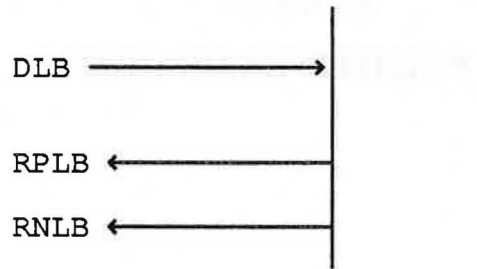
- Primitive de Demande de Sélection des Fiches : DSF
- Primitive de Réponse Positive à la Sélection des Fiches : RPSF
- Primitive de Réponse Négative à la Sélection des Fiches : RNSF

##### 1.2 LECTURE DES FICHES.



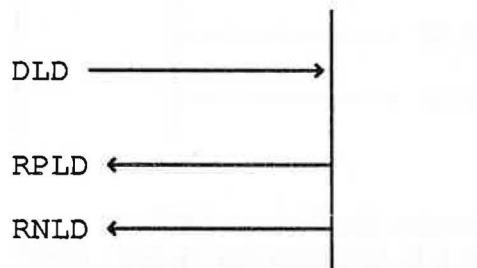
- Primitive de Demande de Lecture d'une Fiche : DLF
- Primitive de Réponse Positive à la Lecture d'une Fiche : RPLF
- Primitive de Réponse Négative à la Lecture d'une Fiche : RNLF

### 1.3 CONSULTATION DE LA LISTE DE BBUA.



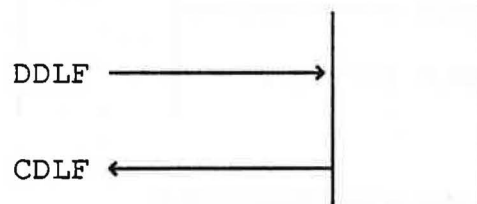
- Primitive de Demande de consultation de Liste des BBUA : DLB
- Primitive de Réponse Positive à la demande de consultation de la Liste des BBUA : RPLB.
- Primitive de Réponse Négative à la demande de consultation de la Liste des BBUA : RNLB

### 1.4 CONSULTATION DE LA LISTE DE DIFFUSION.



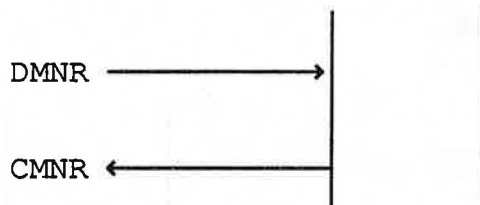
- Primitive de Demande de consultation de la Liste de Diffusion : DLD
- Primitive de Réponse Positive à la demande de consultation de la Liste de Diffusion : RPLD
- Primitive de Réponse Négative à la demande de consultation de la Liste de Diffusion : RNLD

### 1.5 DESTRUCTION LOCALE D'UNE FICHE.



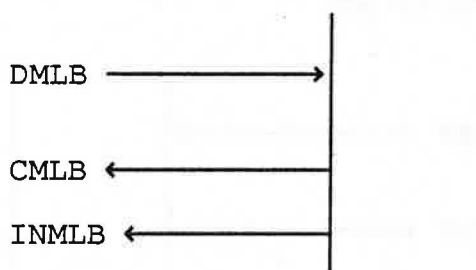
- Primitive de Demande de Destruction Locale d'une Fiche : DDLF
- Primitive de Confirmation de Destruction Locale d'une Fiche : CDLF

## 1.6 MODIFICATION DU NOM DU BBGA DE REFERENCE.



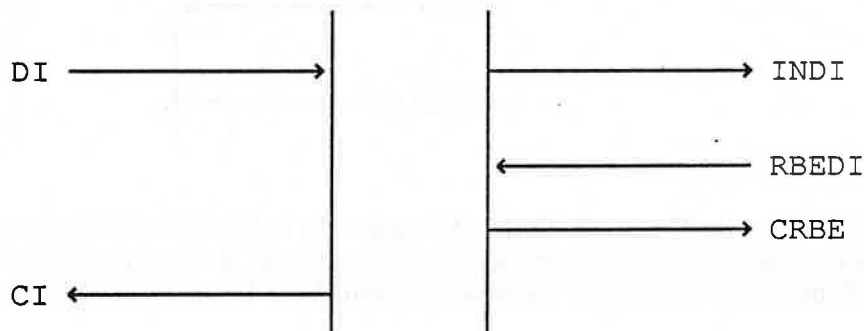
- Primitive de Demande de Modification du Nom du BBGA de Référence : DMNR
- Primitive de Confirmation de Modification du Nom du BBGA de Référence : CMNR

## 1.7 MODIFICATION DE LA LISTE DES BBUA.



- Primitive de Demande de Modification de la Liste des BBUA : DMLB.
- Primitive de Confirmation de Modification de la Liste des BBUA : CMLB
- Primitive d'INDication au BBUA concerné de Modification de la Liste des BBUA : INMLB

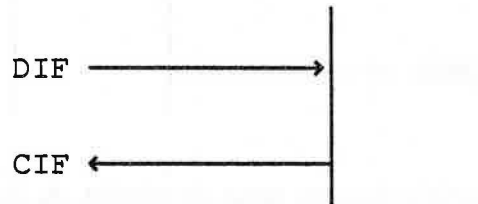
## 1.8 IMPRIMATUR



- Primitive de Demande d'Imprimatur : DI
- Primitive d'INDication d'une Demande d'Imprimatur : INDI
- Primitive de Réponse du BBUA d'Editeur à la Demande d'Imprimatur : RBEDI

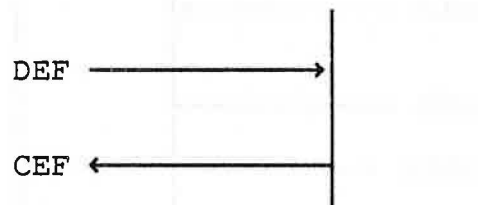
- PrIMITIVE de Confirmation à une Réponse du BBUA d'Editeur : CRBE
- PrIMITIVE de Confirmation d'Imprimatur : CI

### 1.9 INSERTION DE FICHE.



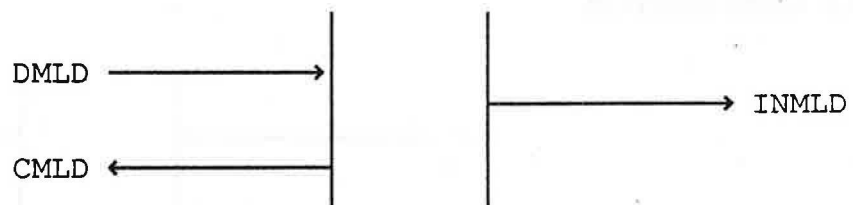
- PrIMITIVE de Demande d'Insertion d'une Fiche : DIF
- PrIMITIVE de Confirmation d'Insertion d'une Fiche : CIF.

### 1.10 EFFACEMENT D'UNE FICHE.



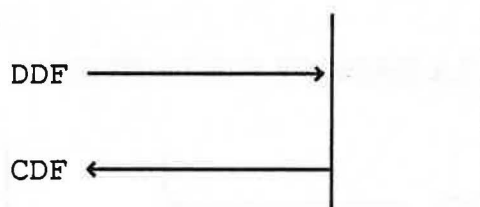
- PrIMITIVE de Demande d'Effacement d'une Fiche : DEF.
- PrIMITIVE de Confirmation d'Effacement de Fiche : CEF

### 1.11 MODIFICATION DE LA LISTE DE DIFFUSION



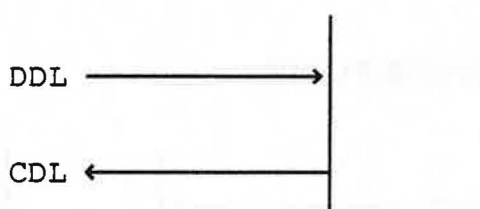
- PrIMITIVE de Demande de Modification de la Liste de Diffusion : DMLD
- PrIMITIVE de Confirmation de Modification de la Liste de Diffusion : CMLD
- PrIMITIVE d'INDication de Modification de la Liste de Diffusion : INMLD

### 1.12 DESTRUCTION D'UNE FICHE.



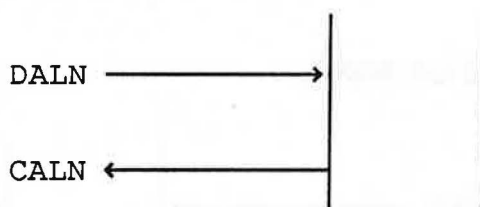
- Primitive de Demande de Destruction d'une Fiche : DDF.
- Primitive de Confirmation de Destruction de Fiche : CDF

### 1.13 DESTRUCTION DE LIEN.



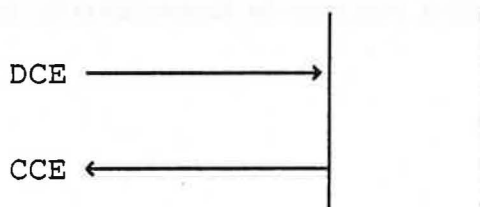
- Primitive de Demande de Destruction de Lien : DDL.
- Primitive de Confirmation de Destruction d'un Lien : CDL

### 1.14 AJOUT D'UN LIEN NEUTRE



- Primitive de Demande d'Ajout d'un Lien Neutre : DALN
- Primitive de Confirmation d'Ajout d'un Lien Neutre : CALN

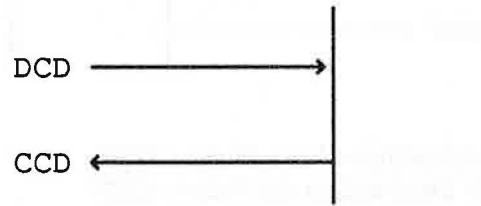
### 1.15 CHANGEMENT DE L'EDITEUR PAR DEFAUT.





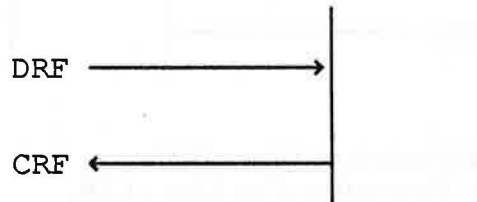
- Primitive de Demande de Changement d'Editeur : DCE.
- Primitive de Confirmation de Changement de l'Editeur : CCE.

### 1.16 CHANGEMENT DE LA DATE DE PEREMPTION.



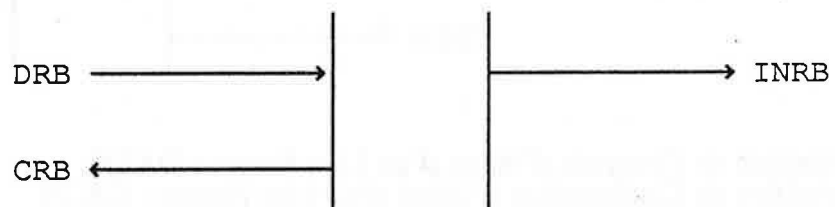
- Primitive de Demande de Changement de Date de péremption : DCD.
- Primitive de Confirmation de Changement de Date de péremption : CCD.

### 1.17 REDISTRIBUTION D'UNE FICHE.



- Primitive de Demande de Redistribution de Fiche : DRF.
- Primitive de Confirmation de Redistribution de Fiche : CRF.

### 1.18 RESTAURATION DE LA BASE.



- Primitive de Demande de Restauration de la BASE : DRB.
- Primitive de Confirmation de Restauration de la Base : CRB.
- Primitive d'INdication au BBUA concerné de Restauration de la Base : INRB.

## **2 PRIMITIVES DE BBUA OFFERTES A L'USAGER**

### **2.1 SELECTION DE FICHES.**

- Primitive de Demande par Usager de Sélection des Fiches : DUSF
- Primitive de Réponse Positive à Usager à sa demande de Sélection de Fiches : RPUSF
- Primitive de Réponse Négative à Usager à sa demande de Sélection des Fiches : RNUSF

### **2.2 LECTURE DES FICHES.**

- Primitive de Demande par Usager de Lecture d'une Fiche : DULF
- Primitive de Réponse Positive à Usager à sa demande de Lecture d'une Fiche : RPULF
- Primitive de Réponse Négative à l'Usager à sa demande de Lecture d'une Fiche : RNULF

### **2.3 CONSULTATION DE LA LISTE DES USAGERS.**

- Primitive de Demande par Usager de consultation de la Liste des BBUA d'Usagers : DULB
- Primitive de Réponse Positive à Usager à sa Demande de consultation de Liste des BBUA d'Usagers : RPUDLB
- Primitive de Réponse Négative à Usager à sa demande de consultation de la Liste des BBUA d'usagers : RNULB

### **2.4 CONSULTATION DE LA LISTE DE DIFFUSION.**

- Primitive de Demande par l'Usager de consultation de la Liste de Diffusion : DULD
- Primitive de Réponse Positive à l'Usager à sa demande de consultation de Liste de Diffusion : RPULD
- Primitive de Réponse Négative à l'Usager à sa demande de consultation de Liste de Diffusion : RNULD

### **2.5 DESTRUCTION LOCALE D'UNE FICHE.**

- Primitive de Demande par AL de Destruction Locale d'une Fiche : DADLF
- Primitive de Confirmation à l'Usager de la Destruction Locale d'une Fiche : CUDLF

### **2.6 MODIFICATION DU NOM DU BBGA DE REFERENCE.**

- Primitive de Demande par l'AL de Modification du Nom du BBGA de Référence : DAMNR
- Primitive de Confirmation à l'Usager de Modification du Nom du BBGA de Référence : CMNR

## 2.7 MODIFICATION DE LA LISTE DES BBUA.

- Primitive de Demande de l'AL de Modification de la Liste des BBUA : DAMLB
- Primitive de Confirmation à l'Usager de Modification de la Liste des BBUA : CUMLB
- Primitive d'INDication à l'utilisateur concerné de Modification de la Liste des BBUA : INUMLB

## 2.8 IMPRIMATUR

- Primitive de Demande par Usager d'Imprimatur : DUI
- Primitive d'INDication à Editeur d'une Demande d'Imprimatur : INEDI
- Primitive de Réponse d'Editeur à la Demande d'Imprimatur : REDI
- Primitive de Confirmation à la Réponse d'Editeur : CER
- Primitive de Confirmation à Usager d'Imprimatur : CUI

## 2.9 INSERTION DE FICHE.

- Primitive de Demande par Usager d'Insertion d'une Fiche : DUIF
- Primitive de Confirmation à Usager d'Insertion d'une Fiche : CUIF.

## 2.10 EFFACEMENT D'UNE FICHE.

- Primitive de Demande par Usager d'Effacement d'une Fiche : DUEF.
- Primitive de Confirmation à Usager d'Effacement de Fiche : CUEF

## 2.11 MODIFICATION DE LA LISTE DE DIFFUSION

- Primitive de Demande par AG de Modification de la Liste de Diffusion : DAMLD
- Primitive de Confirmation à Usager de Modification de la Liste de Diffusion : CUMLD
- Primitive d'INDication à AL de Modification de la Liste de Diffusion : INAMLD

## 2.12 DESTRUCTION D'UNE FICHE.

- Primitive de Demande par AG de Destruction d'une Fiche : DADF.
- Primitive de Confirmation à Usager de Destruction de Fiche : CUDF

### **2.13 DESTRUCTION DE LIEN.**

- Primitive de Demande par AG de Destruction de Lien : DADL.
- Primitive de Confirmation à Usager de Destruction d'un Lien : CUDL

### **2.14 AJOUT D'UN LIEN NEUTRE**

- Primitive de Demande par AG d'Ajout d'un Lien Neutre : DAALN
- Primitive de Confirmation à Usager d'Ajout d'un Lien Neutre : CUALN

### **2.15 MARQUAGE DE FICHE.**

- Primitive de Demande par Utilisateur de Marquage de Fiche : DUMF.
- Primitive de Confirmation à Usager de Marquage de Fiche : CUMF.

### **2.16 RESTAURATION DE LA BASE.**

- Primitive de Demande par AG de Restauration de la Base : DARB.
- Primitive de Confirmation à Usager de Restauration de la Base : CURB.

### **2.17 REDISTRIBUTION D'UNE FICHE.**

- Primitive de Demande par AG de Redistribution d'une Fiche : DARF.
- Primitive de Confirmation à Usager de Redistribution de Fiche : CURF.

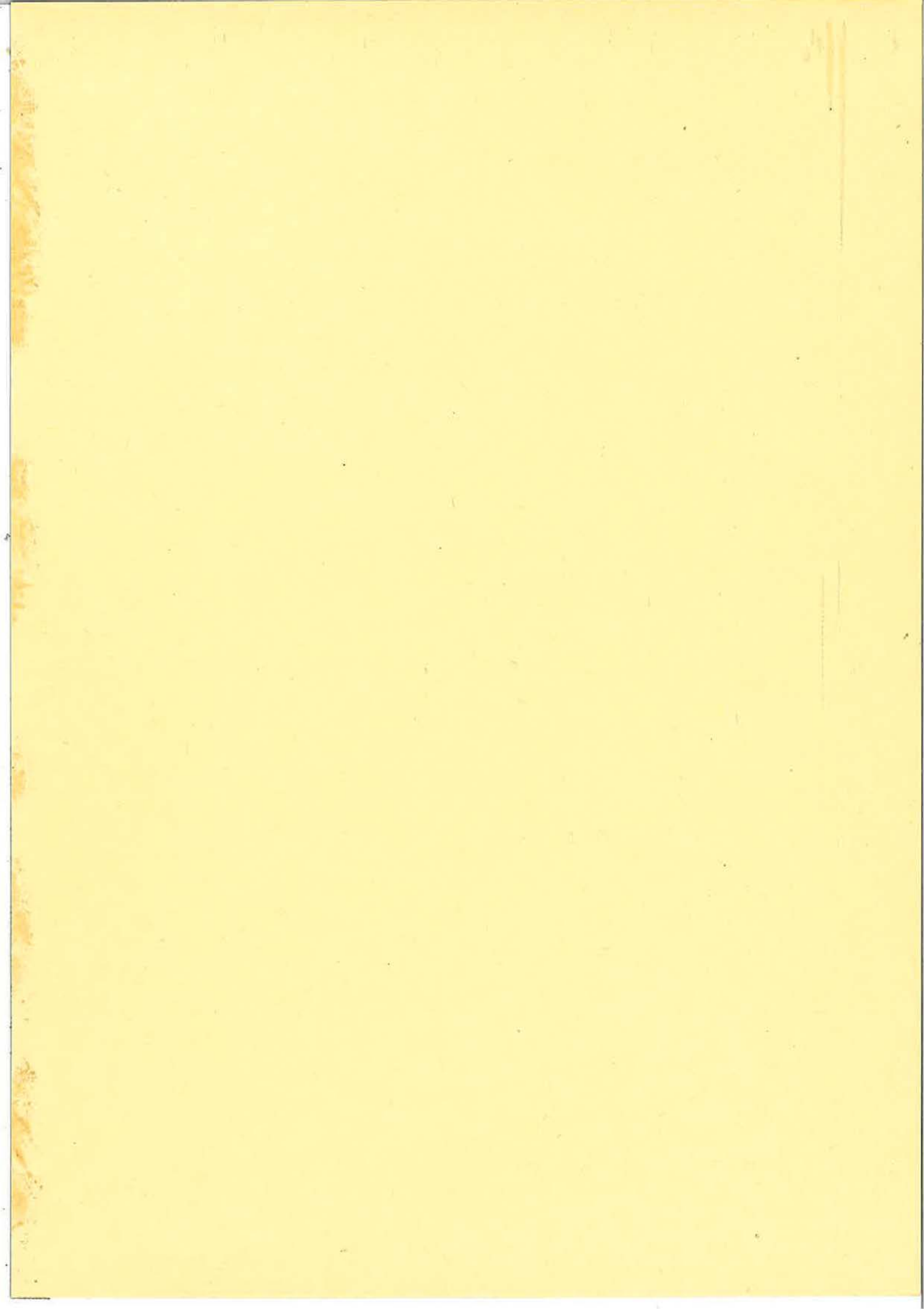
### **2.18 MODIFICATION DE LA DATE DE PEREMPTION**

- Primitive de Demande par AL de Modification de Date de péremption : DAMD.
- Primitive de Confirmation à Usager de Modification de Date de Péremption : CUMDP

### **2.19 MODIFICATION DU NOM D'EDITEUR**

- Primitive de Demande par AL de Modification du nom de l'Editeur : DAME
- Primitive de Confirmation à Usager de Modification du nom de l'Editeur : CUME





## Résumé :

Les conférences informatiques sont promises à un bel avenir. Toutefois, leur essor est freiné par le coût des communications, les difficultés inhérentes à ce type d'outils qui s'adressent à un grand nombre d'utilisateurs, mettent en jeu de grandes quantités d'informations et couvrent un large champ d'applications, et par la nécessité d'offrir aux usagers un accès simple, confortable et efficace.

La solution que nous proposons répond à ces problèmes par les options suivantes. L'ensemble des messages échangés est archivé dans une base structurée ce qui permet de les gérer efficacement. Cette base est dupliquée sur différents sites pour réduire les coûts de consultation. Les communications entre sites se font en mode messagerie telle qu'elle est recommandée par le CCITT. Pour une bonne coopération des usagers, des rôles sont définis, des droits leur sont associés. Enfin, l'interface avec l'utilisateur tire parti des progrès réalisés dans ce domaine tant dans le matériel que dans les logiciels actuellement disponibles.

## Mots clés :

*Informatique*

— Conférence, Messagerie, X400, Communication de groupe, Interface utilisateur