

# Développement d'un Gestionnaire Automatisé de Bibliographies selon les normes de l'*American Psychological Association* (*GABI APA Style*)

Paul THIRION

## Abstract

Nowadays, there are plenty of complete and user-friendly database management systems. However, most of them only produce poorly organized raw data, generally lacking of internal consistency. This situation remains awkward, especially for scientists who have regularly to report references in an extremely ruled international format. *GABI APA Style* was built in this context.

*GABI APA Style* is a reference manager software using the presentation rules of the American Psychological Association (A.P.A.) It was written in *Clipper* language conforming to the relational database principle. The initial aim was to elaborate a software to centralize references, filling in the intellectual background of the Department of Psychology and Educational Sciences of Liège University.

Special attention was paid to the ergonomic and easy-to-use aspects of the software, just as the possibilities of transfers with other applications.

---

✉ Université de Liège; Faculté de Psychologie et Sciences de l'Éducation; U.D. Psychologie;  
B-4000 Sart Tilman; Liège (Belgique).

Fax : + 32 41 56 29 44

E-mail : u0020a1@bliulg11

---

MOTS-CLÉS : logiciel-développement, bases de données relationnelles, ergonomie logicielle, références bibliographiques

---

## 1. Choix initiaux

### 1.1. Choix des normes de l'A.P.A.

Aucune norme bibliographique n'est universelle. Qui plus est, toutes sont entachées de limites désagréables. En choisir une a toujours quelque chose de restrictif et reste sujet à controverse. Aussi, les éléments qui ont présidé au choix des normes de l'*American Psychological Association* pour développer *GABI* sont-ils purement d'ordre pragmatique :

- au niveau des utilisateurs cibles<sup>1</sup>, elles étaient déjà répandues;
- au niveau international, ce sont celles qui sont le plus largement utilisées dans le domaine de la psychologie.

Les descriptions ISBD (*International Standard Bibliographic Description*), dont l'usage est généralisé dans les bibliothèques, auraient eu l'avantage de dépasser le cadre limité de la psychologie et d'être plus complètes et précises que les normes de citation de références. Cependant, leur objectif est la description catalographique du document par le bibliothécaire. Par conséquent, elles sont encombrées de détails inutiles pour l'utilisateur final. Elles sont aussi plus limitées dans leurs possibilités d'exploitation concrète par le scientifique, leur usage effectif ne dépassant guère le cadre des bibliothèques. De plus, à un niveau plus épistémologique, l'ISBD aborde le document en tant qu'objet matériel, tandis que les normes de citation de références bibliographiques le considèrent dans son contenu, en tant que thématique développée par un auteur, position beaucoup plus proche des besoins et des préoccupations du chercheur.

Les règles de l'A.P.A. ont été élaborées pour standardiser la présentation des références dans les publications de l'association. Elles se caractérisent par leur souci d'efficacité : leur seul objectif est de permettre l'identification rapide et univoque du document. Elles éliminent par conséquent tout détail inutile dans ce cadre.

Elles sont décrites en une vingtaine de pages dans un ouvrage précisant l'ensemble des règles stylistiques exigées des auteurs proposant une publication à l'A.P.A. (*American Psychological Association*, 1984). Malheureusement, elles manquent d'exhaustivité, un certain nombre de cas particuliers restant non résolus.

Enfin, comme toute règle, elles ont quelque chose de contraignant et de rigide dont il faut s'accomoder et leur américano-centrisme est parfois gênant.

---

<sup>1</sup> Membres de la Faculté de Psychologie et des Sciences de l'Éducation de l'Université de Liège.

## 1.2. Nécessité de développement d'une application spécifique

Un tour d'horizon des logiciels existants a montré l'absence ou l'inadéquation de produits permettant de répondre aux objectifs fixés, rendant impossible l'économie du développement d'une application spécifique.

### 1.2.1. Inadéquation de *LIBER*<sup>2</sup>

Il n'était pas judicieux d'utiliser *LIBER* dans le contexte particulier de la gestion de références bibliographiques de scientifiques pour plusieurs raisons :

- Les finalités sont différentes. *LIBER* a été installé pour gérer les documents *effectivement* possédés par les bibliothèques. Or, l'objectif poursuivi ici était de rassembler l'information sur des publications, qu'elles soient ou non disponibles.
- La politique du réseau des bibliothèques et U.D. (Unités de Documentation) de l'Université est d'introduire dans *LIBER* les ouvrages entiers, les thèses et les titres des collections de périodiques. Y introduire des articles de périodiques, des chapitres d'ouvrages, voire des rapports de recherche non publiés ... en dehors d'une politique systématique, aurait eu pour conséquence d'encombrer et rendre hétéroclite la base de données actuelle.
- *LIBER* ne permet pas de formater les références selon les normes de l'APA. S'il eut sans doute été possible de développer sous *PICK*<sup>3</sup> des modules permettant ce type de formatage, l'investissement en programmation n'aurait sans doute pas été plus faible que pour un développement spécifique et ce sous-produit à usage d'une seule Faculté aurait encombré le système pour les autres utilisateurs.
- L'exportation sous *LIBER* ne peut se faire que via la médiation des gestionnaires du réseau. Or, pour être *effectivement* utilisé par les scientifiques, il était nécessaire de disposer d'un outil ouvert, dont les produits puissent être immédiatement récupérés par l'utilisateur final pour d'autres applications (traitement de texte, ...).
- Enfin, les critères de recherche disponibles avec *LIBER* semblaient trop limités dans ce contexte spécifique.

---

<sup>2</sup> *LIBER* (Logiciel Intégré pour Bibliothèques Et Recherches documentaires) est le logiciel bibliothéconomique en cours d'implantation dans le réseau des bibliothèques et unités de documentation de l'Université de Liège.

<sup>3</sup> *PICK* est le système d'exploitation dans lequel fonctionne *LIBER*.

### 1.2.2. Autres produits exploitant les normes bibliographiques de l'A.P.A.

Lorsque la décision de développer *GABI* a été prise, le seul logiciel existant sur le marché permettant de gérer des références selon les normes de l'A.P.A. était le traitement de texte *Manuscript Manager APA Style™* de *Pergamon Press®* qui prend en compte toutes les règles éditoriales de l'A.P.A.<sup>4</sup> Cependant, il présente un certain nombre de limites :

- Il nécessite une connaissance préalable des normes de l'A.P.A. ;
- Le cadre est celui d'un traitement de texte et non d'un outil de gestion de bases de données. Bien qu'il soit possible de créer des bibliothèques de références, elles n'offrent aucune possibilité de recherche, sinon le balayage exhaustif ;
- Le système a été conçu pour un utilisateur individuel ayant un nombre limité de références à gérer. Il est difficile d'imaginer son comportement face à quelques centaines voire quelques milliers de références ;
- Alors que les possibilités de traitement de texte sont assez élémentaires, il s'agit d'un système fermé ne permettant pas d'échange de données avec l'environnement.

Un autre outil permettant de gérer des références selon des normes établies est le *Reference Manager®* développé par *Research Information Systems™*. Ce logiciel disponible sur *Macintosh®* et sur *PC* présente de nombreuses fonctionnalités communes avec *GABI*, mais aussi des potentialités supplémentaires intéressantes ainsi que des désavantages non négligeables.

#### Avantages

- Possibilité d'élaborer soi-même des formats d'affichage en paramétrant l'ordre et la présentation des éléments. (Ce logiciel ne gère pas une norme particulière, mais se veut adaptable à toutes) ;
- Possibilité de créer des liens dynamiques avec certains traitements de texte. L'avantage de cette technique par rapport aux exportations / incorporations de *GABI*, est que les modifications apportées dans la base de données se répercutent immédiatement dans les textes qui y sont liés et inversement ;
- Enfin, il est d'ores et déjà disponible en version *Macintosh* et *PC* compatible.

---

<sup>4</sup> Ce logiciel a d'ailleurs été reconnu par l'A.P.A. comme le logiciel officiel de cette association pour la préparation de textes respectant les règles de style de l'A.P.A.

### Inconvénients par rapport à *GABI APA Style*

- Il nécessite une connaissance préalable approfondie des normes pour permettre la paramétrisation;
- Seuls les cas simples sont traités (ouvrages, articles de périodiques, chapitres d'ouvrages, ...). Rien n'est prévu pour les thèses de doctorat, rapports de recherche, actes de congrès, ...;
- Même dans ces types de documents, la paramétrisation ne permet pas de gérer de manière satisfaisante les nombreux cas particuliers;
- Les contrôles automatiques à la saisie sont limités, permettant l'introduction d'incohérences. L'assistance au moyen des fichiers d'autorité n'est pas automatique;
- Les seules informations complémentaires possibles sont des descripteurs et il n'y a pas de création automatique d'un dictionnaire de mots-clés;
- Les critères de recherche sont limités (auteur, année, titre de périodique, descripteur);
- Quelques essais avec un *Macintosh* puissant et sur une base limitée à une septantaine de références semblent indiquer des procédures de recherche peu rapides. Il n'est pas certain que ce logiciel puisse supporter des volumes beaucoup plus élevés, en conservant des délais d'attente acceptables;
- Il n'existe pas de possibilité d'importation/exportation sur disque des références selon le standard ISO 2709.

## 1.3. Choix de *Clipper*<sup>™</sup>

### 1.3.1. Caractéristiques et avantages

*Clipper*<sup>™</sup>, produit par *Nantucket Corporation*<sup>®</sup>, s'est fait connaître dès 1985 comme un compilateur extrêmement performant du *dBase*<sup>™</sup> d'*Ashton-Tate*<sup>®</sup>.

Au-delà de cette fonction, il s'agit aussi d'un véritable environnement de programmation orienté bases de données, se démarquant de plus en plus du *dBase*. S'il utilise le contexte *dBase* pour la structure et la gestion des fichiers, la philosophie de programmation, la définition de variables..., il offre des améliorations considérables (gestion de l'environnement, manipulation de tableaux, index, nombre de fichiers pouvant être ouverts simultanément, ...). De plus, outre la palette de commandes et fonctions *dBase*, il en ajoute en standard beaucoup d'autres, extrêmement puissantes, qui en font un langage de haut niveau.

Il permet aussi d'exploiter de très nombreuses et très complètes bibliothèques de fonctions spécialisées, (en particulier *Clipper Tools One*<sup>™</sup> de *Nantucket* largement utilisé dans le développement de *GABI*). Enfin, il offre la

possibilité de créer ses propres commandes et fonctions en langage C et de les incorporer dans le code. Avec ces outils complémentaires, il permet de sortir du cadre des bases de données pour développer tout autre type d'applications.

### 1.3.2. Principales limites

Une limite de *Clipper* concerne la taille très importante des fichiers exécutables qu'il crée et qui s'explique par sa philosophie de compilation (intégration de l'ensemble des ressources dans le module exécutable, contrairement à d'autres compilateurs qui ne traduisent que les commandes et fonctions réellement utilisées dans le programme source<sup>5</sup>).

Malgré cela, même la dernière version de *Clipper* ne permet pas d'utiliser la mémoire étendue du système. La seule forme de mémoire non conventionnelle exploitable avec cette version 5 est la mémoire expansée de type LIM (*Lotus-Intel-Microsoft*), nettement moins répandue. De plus, sa gestion ne peut être que difficilement modulée par le développeur. Il est dès lors nécessaire, pour développer des applications importantes en *Clipper* version *summer 87*, de réduire la taille du programme en mémoire en s'attelant à la création d'*overlays*.

Il faut espérer que l'ouverture amorcée dans la version 5 vers les compléments de mémoire s'accroîtra et que les versions ultérieures permettront d'accéder à la zone de mémoire haute et étendue.

Un autre inconvénient de *Clipper* est une conséquence du précédent. L'opération de compilation, et surtout celle de «*link*» est très longue, ce qui est pénible en phase de débogage.

Enfin, *Clipper* est prévu pour fonctionner en mode d'affichage texte. Pour avoir accès au mode graphique, il faut utiliser des bibliothèques de fonctions supplémentaires, ce qui agrandit encore la taille du module exécutable et limite l'accès à certaines fonctions conçues pour le mode texte.

### 1.3.3. Choix de la version «*Summer 87*»

- Trois raisons expliquent le choix de cette version plutôt que la version 5 :
- Le développement de *GABI* a duré près de 2 ans. Lorsque début 1992, *Clipper 5* a fait son apparition, l'avancement de *GABI* était tel qu'une mise à jour aurait occasionné un retard important;

---

<sup>5</sup> Ainsi, la compilation avec *Clipper* d'un pseudo-programme d'une seule ligne de code : un nettoyage d'écran par exemple, a comme résultat un module exécutable de 157 Ko!

- La compatibilité totale annoncée avec la version précédente n'est pas valable pour la bibliothèque *Clipper Tools One*, pourtant produite par *Nantucket* également;
- La possibilité de créer des *overlays* dynamiques qui aurait pu être intéressant pour le développement de *GABI*. Cependant, l'expérimentation montre qu'ils ralentissent l'exécution du programme par rapport à une structure d'*overlays* statiques bien conçus.

Quoi qu'il en soit, il est certain que les développements futurs de *GABI*, demanderont l'exploitation des potentialités nouvelles qu'offriront les mises à jour ultérieures de *Clipper*, et des bibliothèques de fonctions qui se développeront dans cet environnement.

## 1.4. Éléments de la référence

### 1.4.1. Aménagements par rapport aux normes A.P.A.

#### 1.4.1.1. Gestion des cas particuliers et exceptions

Vouloir inclure systématiquement toutes les exceptions des normes A.P.A. dans *GABI*, quel que soit leur degré d'occurrence, aurait encombré inutilement le logiciel. Aussi, les règles ne présentant pas une probabilité minimum d'apparition dans le contexte visé ont-elles été négligées. Le cas échéant, l'utilisateur adaptera aisément sa référence à une des possibilités prévues.

Les principaux cas non traités actuellement par *GABI* sont :

- citation d'un abstract et non de la référence complète;
- lettre à un éditeur;
- thèse de Master;
- microfilm d'une thèse de doctorat;
- manuscrit soumis pour publication mais non encore accepté;
- interview;
- documents non publiés à l'exception des thèses de doctorat et des mémoires de licence;
- support non imprimé : film, enregistrement audio, logiciel informatique, ...;
- article de magazine ou quotidien;
- « review » d'article.

S'il s'avère que certains cas non traités se présentent de manière significative, ce choix empirique sera revu. Ainsi, il apparaît dès à présent que les deux derniers devront être incorporés à la prochaine version du logiciel.

#### 1.4.1.2. Aménagements formels

Afin d'éviter un américano-centrisme trop poussé, *GABI* ne respecte pas la règle de faire suivre le titre original d'un ouvrage publié dans une langue autre que l'anglais par sa traduction en anglais entre crochets. Libre à chacun de l'ajouter lui-même à la saisie.

De même, pour les thèses de doctorat non publiées, la mention « *unpublished doctoral dissertation* » a été remplacée par son équivalent français auquel a été ajouté le domaine, non exigé par l'A.P.A. (par exemple : « *thèse de doctorat en psychologie non publiée* »).

Enfin, bien que les normes de l'A.P.A. soient muettes sur les mémoires de licence, ce type de document fréquemment exploité dans nos universités a été ajouté sur le modèle de la présentation des thèses de doctorat.

#### 1.4.2. Informations complémentaires

En plus des éléments de la référence A.P.A., des informations complémentaires ont été ajoutées. Elles permettent de dépasser la simple citation bibliographique et, par l'analyse qu'ils autorisent, enrichir les possibilités d'exploitation de la base de données :

- le *service* au sein duquel le document a été élaboré;
- la *langue* du document;
- une *cote de rangement*;
- jusqu'à 20 *descripteurs*.

À ces éléments optionnels, *GABI* ajoute :

- l'*identificateur de la base* : suite de 6 caractères attribuée lors de la création de la base en cours et qui correspond à son identité;
- le *numéro d'identification de référence* : numérotation monotone attribuée par le système lors de la création de la référence et qui constitue sa clé d'accès. Joint à l'identificateur de la base, il devient l'étiquette perpétuelle de la référence;
- son *étiquette d'origine*, si la référence a été importée d'une autre base;
- la *date de création* de la référence;
- le cas échéant, sa *date de modification*.

Selon le souhait de l'utilisateur, ces éléments complémentaires peuvent ou non apparaître sur l'écran et à l'impression.



Exemple de référence produite par *GABI*

Richelle, M. (1987e). Variation and selection: The evolutionary analogy in Skinner's theory. In S. Modgils & C. Modgils (Eds.), *B.F. Skinner: Consensus and controversy* (pp. 127-137). New York, NY: Falmer Press.

NUMREF : UDPSY 213      CREE LE 01/07/1991      MODIFIE LE : 04/05/1992  
 ORIGINE : FAPSE 213      LANGUE : Anglais  
 Service de Psychologie Expérimentale

COMPORTEMENT      REVUE CRITIQUE      VARIABILITE COMPORTEMENTALE

## 2. Développement du logiciel

Le développement de *GABI* s'est organisé autour de quatre axes généraux :

- Être *complet et performant*, pour permettre de gérer avec rapidité, plusieurs milliers de références et offrir également un maximum d'autres potentialités d'exploitation. Pour cela, une attention particulière devait être accordée à la *structure des fichiers et index*.
- Être *convivial*, agréable à utiliser, et accessible à quiconque, sans connaissances informatiques ou bibliographiques préalables. Cela impliquait un travail important au niveau de *l'ergonomie* du logiciel et rendait impossible de faire l'économie de la gestion des erreurs.
- Être utilisable dans la majorité des environnements *PC* compatibles. Ceci nécessitait, un travail important sur *l'utilisation de la mémoire vive*, étant donné la taille de l'application que supposaient les deux premiers axes.
- Enfin, être ouvert et permettre de *communiquer* avec d'autres applications : des traitements de texte, mais aussi des systèmes de gestion de bases de données.

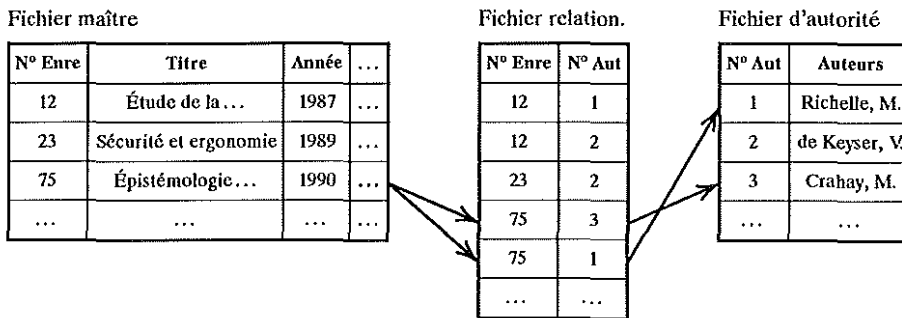
### 2.1. Structure des fichiers et index

Le principe directeur de performance du logiciel<sup>6</sup> impliquait une structuration complexe des fichiers. Avec les fichiers d'autorité, les fichiers relationnels et les index et sans compter les fichiers temporaires de travail, *GABI* fonctionne avec près de 70 fichiers ouverts simultanément.

<sup>6</sup> Le concept de performance et par là de « rapidité » doit être compris à un niveau subjectif : c'est le délai que la plupart des utilisateurs sont prêts à accepter pour une opération. Dans la pratique, à l'heure actuelle, excepté pour les gros traitements, ce délai ne peut dépasser la seconde.

### 2.1.1. Structure relationnelle

*GABI* a été développé selon le principe des bases de données relationnelles. Le principe en est très différent des bases de données classiques. La référence est éclatée en de multiples éléments, enregistrés séparément. Pour chaque type d'information présentant un caractère répétitif d'une référence à l'autre (auteurs, ...) et pour lequel une possibilité de vérification à la saisie est souhaitée, une base séparée, un *fichier d'autorité* est créé. La base principale ne comporte plus alors que les éléments non répétitifs. Dans chaque fichier d'autorité, les éléments ne sont introduits qu'une seule fois, ce qui réduit la redondance et les problèmes de graphies multiples. Entre la base principale (*fichier maître*) et chaque fichier d'autorité, une base «*relationnelle*» est créée pour spécifier quels éléments du fichier d'autorité sont liés à un enregistrement du fichier maître.



Cette structure relationnelle, outre la possibilité de contrôle de la saisie qu'elle permet, rentabilise plus efficacement l'espace disque (gain de place estimé pour *GABI* : de 60 à 80 %). Elle est cependant plus lourde à concevoir et à gérer car chaque présentation de l'enregistrement, suppose sa reconstitution à partir des différents fichiers d'autorité. De même, les mises à jour et modifications doivent être répercutées sur l'ensemble des fichiers concernés.

Dans *GABI*, à côté du fichier maître, 9 fichiers d'autorité ont été construits, munis chacun d'un fichier relationnel. Les trois premiers sont communs à l'ensemble des références : *auteurs*, *descripteurs* et *mots-clés*. Les quatre suivants sont spécifiques à certains types de documents : *villes* et *pays* de publication, *éditeurs* commerciaux, *titres de périodiques* et *universités* de défense des thèses ou mémoires. Enfin, les deux derniers ont un statut particulier : liste des *pays connus* (utilisée uniquement pour faciliter la saisie); et *antidictionnaire multilingue* (voir *infra*, section 2.1.4).

## 2.1.2. Champs

### 2.1.2.1. Champs à longueur fixe et champs à longueur variable

*Clipper* est conçu pour fonctionner avec des *champs de longueur fixe* occasionnant une perte de place inévitable, mais permettant l'utilisation d'index très performants.

Cependant, un certain nombre d'éléments du fichier maître ne nécessitant pas d'indexation ni de recherches spécifiques, il leur a été appliqué la technique des *champs à longueur variable* particulièrement économique en espace disque. Certains sont communs à toutes les références (titre et sous-titre), d'autres sont particuliers à certains types seulement (pagination pour les périodiques et articles d'ouvrages avec éditeurs scientifiques, ...).

Pour optimiser encore l'utilisation de l'espace disque, ces éléments ont été séparés par des identificateurs et concaténés avant d'être intégrés en un seul champ « mémo ».

### 2.1.2.2. Accepter une certaine redondance

L'évitement de toute redondance n'était pas pertinent dans le cas de *GABI*. En effet, les normes de l'A.P.A. exigent un classement des références par ordre alphabétique du champ auteurs dans sa globalité, un sous-classement sur la date de publication puis sur le titre le cas échéant. Les auteurs étant repris isolément dans un fichier d'autorité séparé, il était nécessaire d'ajouter au fichier maître, une « image » de la rubrique « auteurs » permettant l'indexation. Les différents auteurs y sont repris dans leur ordre d'apparition, en lettres capitales pour éviter les problèmes de classement sur les caractères accentués. À l'exception du premier auteur, les initiales des prénoms ont été éliminées, le risque d'ambiguïté de classement à ce niveau étant négligeable. Dans le même ordre d'idée, la longueur totale du champ a été limitée pour permettre d'y inclure une dizaine d'auteurs.

Par contre, il est envisageable de rencontrer deux références des mêmes auteurs, publiées la même année, ce qui impose le classement sur le titre. Les champs « mémo » ne permettant pas d'indexation, il a fallu ajouter un champ à longueur fixe reprenant les premiers caractères du titre du document. Si le nombre limité de caractères s'avère insuffisant (par exemple : un même document publié en deux parties, distinctes par le sous-titre), il fera l'objet d'un réaménagement ultérieur.

Enfin, cette image du début du titre a subi une double transformation : typographie pauvre pour éviter les problèmes de classement et suppression de l'article initial éventuel.

### 2.1.3. Index préétablis

Il eut été possible de développer des structures de fichiers inversés semblables à celles qu'utilise *STAIRS*<sup>®</sup> par exemple. Cette technique permet des recherches extrêmement rapides. Par contre, la lourdeur des opérations de mise à jour rend celles-ci inimaginables en temps réel. Ceci implique le maintien en parallèle d'une base de données effective et de fichiers de mise à jour, et donc l'impossibilité d'exploiter les références introduites depuis la dernière opération de mise à jour. De plus, et c'est une conséquence, elle implique une consommation-disque très importante.

Pour ces raisons, les index classiques permettant des mises à jour en temps réel, ont été préférés. Dans ce cadre, deux stratégies générales étaient envisageables :

- index *a posteriori* : ne définir au départ que les index strictement nécessaires au fonctionnement du système et construire les autres, chaque fois que nécessaire;
- index *a priori* : construire au départ l'ensemble des index susceptibles d'être utilisés et les mettre à jour en temps réel.

Cette seconde politique a été adoptée. Elle occasionne un coût en espace disque nettement plus élevé et nécessite l'ouverture simultanée d'un plus grand nombre de fichiers (48 fichiers d'index différents), ce qui augmente les besoins en mémoire vive. Par contre, elle permet d'éviter tout délai lors du passage d'un type de classement à l'autre, offrant ainsi un meilleur confort d'utilisation<sup>7</sup>, d'autant que le ralentissement à la saisie dû à la mise à jour des index est quasi insensible.

Dans le but d'optimiser vitesse et efficacité, ce principe souffre une exception. Lorsque le nombre de références à communiquer est restreint, le balayage systématique du fichier maître dans l'ordre de l'index choisi, pour repérer les numéros de références sélectionnés, occasionnerait des délais inacceptables. *GABI* construit alors un fichier temporaire, indexé selon la clé choisie et y reproduit les références nécessaires à partir du fichier maître. Cette technique améliore sensiblement les délais dès qu'il y a moins de 10 % de références à communiquer.

### 2.1.4. Fichier des mots-clés et antidictionnaire multilingue

Une des fonctionnalités de *GABI* est de permettre la recherche de références à partir de mots-clés (n'importe quel mot significatif de la référence ou

---

<sup>7</sup> La création d'un index portant sur plusieurs centaines d'enregistrements peut durer plusieurs minutes.

du champ descripteur). Lors de la saisie d'une référence, ceux-ci sont repérés et ajoutés automatiquement à un fichier *dictionnaire*. Pour éliminer les mots non-significatifs qui l'alourdiraient inutilement, 3 techniques sont utilisées :

- élimination des mots de moins de 2 lettres (rarement sémantiquement forts);
- omission des groupes de caractères non lexicaux (pagination, numéro de volume);
- confrontation avec un *antidictionnaire multilingue*.

Il n'a pas paru intéressant de développer un outil syntaxique permettant d'éliminer les variations flexionnelles. Dans la plupart des cas, l'utilisation de la troncature y remédie aisément.

L'*antidictionnaire* est constitué sous forme d'une base de données des mots non significatifs les plus fréquents et de leurs dérivations syntagmatiques.

Pour certains mots sans signification dans une langue, il existe un homographe porteur de sens dans une autre langue (la préposition néerlandaise *met* et le verbe français *mettre* à la troisième personne de l'indicatif présent; la préposition allemande *wegen* qui «à cause de» et le verbe néerlandais *wegen* : «peser»; le pronom relatif allemand *deren* et le verbe néerlandais *deren* «nuire»; ...). Pour cette raison, l'*antidictionnaire* a été rendu multilingue : ne seront éliminés que les mots sans signification de la langue d'appartenance du document.

Ceci ne résout pas le problème des homographies au sein d'une même langue (en français : la conjonction et le substantif *car*; la préposition et le substantif polysémique *vers*; la préposition et le substantif pluriel vieilli *sous*, ...). Dans de tels cas, la décision d'inclusion du terme dans l'*antidictionnaire* est basée sur un principe pragmatique. Si l'acception porteuse de signification a des chances d'être rencontrée plus qu'exceptionnellement dans le contexte de la psychologie et des sciences de l'éducation, le terme n'est pas inclus dans l'*antidictionnaire*. Il est préférable d'encombrer le dictionnaire de quelques termes sans signification, plutôt que de perdre la possibilité de faire des recherches sur un terme prometteur.

Ce dictionnaire négatif multilingue comporte au départ 1 100 mots appartenant aux 7 langues les plus couramment employées par les utilisateurs cibles (français, anglais, néerlandais, allemand, italien, espagnol et portugais). Il est possible de le compléter avec d'autres termes issus d'une des 11 langues connues de *GABI*, via le module de gestion des fichiers d'autorité.

Avec leurs index, les fichiers dictionnaires constituent les plus gros fichiers mis en place par *GABI*. Ceci ne pose aucun problème pour le fichier dictionnaire

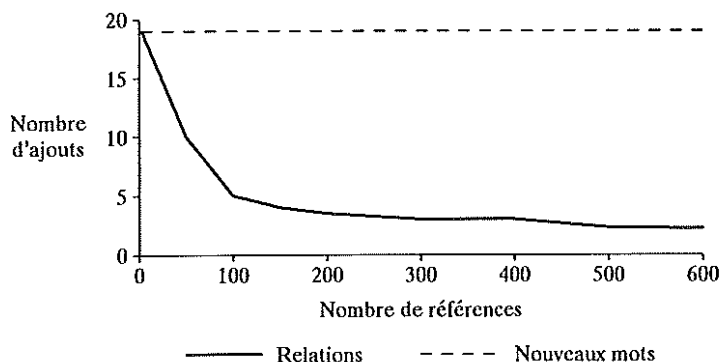


Fig. 1.- Accroissement des fichiers dictionnaire et relationnel en fonction du nombre de références introduites.

lui-même dont la croissance suit une courbe logarithmique<sup>8</sup>, par contre, la taille du fichier relationnel et de ses index augmente de manière linéaire (en moyenne 19 mots significatifs traités par notice). Outre l'espace disque que cela nécessite<sup>9</sup>, l'importance de ces fichiers est telle qu'avec un système disposant de mémoire vive limitée, une opération lourde telle que la réorganisation des fichiers pourrait être entravée.

## 2.2. Convivialité et ergonomie logicielle

Le deuxième souci majeur qui a présidé au développement de *GABI* était la volonté de créer un outil agréable et facile à utiliser par tout utilisateur. Ceci impliquait un travail au niveau de la couche superficielle de l'interface utilisateur, mais aussi à un niveau plus profond.

### 2.2.1. Niveau superficiel

#### 2.2.1.1. Gestion des écrans

Une attention particulière a été consacrée à la construction des écrans pour permettre une lecture aisée et un repérage rapide de l'information. Plusieurs techniques ont été utilisées :

<sup>8</sup> La première référence implique la création d'environ 19 mots dans le dictionnaire, mais, lorsque 600 références ont été incorporées. Les suivantes n'ajoutent plus en moyenne que 2,2 mots nouveaux, ce qui est une démonstration manifeste de l'intérêt des structures relationnelles.

<sup>9</sup> Avec une base de données de 1 800 références, chaque index du fichier relationnel occupe plus d'un demi méga-octet d'espace disque. Cette taille peut même dépasser le méga-octet si la base n'a pas subi récemment une réorganisation.

- *multi-fenêtrage* : l'affichage est organisé autour d'un écran de base sur lequel s'ouvrent en superposition, en relief et éventuellement en cascade, différentes fenêtres, en fonction de l'opération effectuée. Ce système renforce l'impression de continuité et aide l'utilisateur à suivre le fil de sa propre démarche à travers la « trace » des écrans précédents;
- *menus déroulants* : cette technique fonctionne de pair avec le multi-fenêtrage. Sur l'écran de base ne sont repris que les titres des grandes familles de fonctions. Les menus complets « se déroulent » à la demande;
- *structuration de l'écran en zones fonctionnelles* : l'affichage d'informations de même ordre se fait toujours dans la même zone de l'écran :
  - la ligne supérieure constitue la « barre-menu principale », réservée aux grandes familles de fonctions de *GABI* ;
  - dans la partie supérieure de l'écran s'affichent les *sous-menus principaux* et les *messages* d'aide ou d'erreur;
  - dans la moitié inférieure apparaîtront les *résultats* (de recherche ou des opérations réalisées par le système, tel qu'une étape d'installation, ...);
  - la dernière ligne de l'écran ou des fenêtres est réservée au *rappel des touches actives*.
- *standardisation des attributs d'écran* : quelle que soit la procédure, les mêmes couleurs sont utilisées pour des opérations ou informations sémantiquement similaires. Ce complément d'information contextuelle vise à faciliter et accélérer l'interprétation du message.

### 2.2.1.2. Assistance à la saisie

- *Simplification de la saisie*

L'utilisateur n'a pas à se préoccuper de la formalisation de la référence selon les règles de l'A.P.A. En cours d'introduction d'une nouvelle référence, *GABI* ne présente que les éléments pertinents pour le type de document spécifié et gère de manière transparente l'ajout de mentions spécifiques, l'ordre des éléments, la ponctuation, les parenthèses, ...

- *Utilisation des listes d'autorité*

Chaque fois que l'utilisateur est amené à introduire un terme contrôlé par un fichier d'autorité, une triple vérification s'opère en cascade :

- 1) existence dans le fichier d'autorité, du terme sous la forme introduite;
- 2) existence d'enregistrements comportant comme racine l'élément introduit;
- 3) existence d'enregistrements ayant au moins les premières lettres communes avec cet élément.

Si ces vérifications ont échoué, *GABI* propose d'ajouter l'élément inconnu, ou de le réintroduire sous une forme correcte.

Dans les cas n° 2 (l'utilisateur se contente d'introduire les premières lettres de l'élément) et n° 3 (erreur de frappe), *GABI* propose une liste des possibilités les plus proches dans laquelle il suffit de choisir l'item adéquat. Cette aide accélère sensiblement la saisie et limite les erreurs de frappe.

- *Auto-corrections et vérification de la cohérence interne*

La *capitalisation* des mots dans les normes de l'A.P.A. suit des règles complexes et d'une logique toute relative. Pour éliminer les risques d'erreurs, *GABI* accepte toutes les formes à la saisie et les adapte automatiquement pour les rendre conformes à la norme.

Il vérifie aussi la présence d'éléments nécessaires et les ajoute en cas d'omission par l'utilisateur (le terme « université » par exemple, dans le cas de l'introduction d'un mémoire ou une thèse).

Enfin, *GABI* contrôle la cohérence de certaines données introduites par l'utilisateur (refus d'une date de publication postérieure à l'année en cours, ou d'une page de fin d'article inférieure à la page de début, impossibilité d'introduction d'une thèse ou d'un mémoire anonyme, ...).

### 2.2.1.3. Aide en ligne « context specific »

Outre les messages d'information élémentaires qui apparaissent lors des différentes opérations, il existe un système d'aide détaillée sur l'utilisation du logiciel, accessible par deux biais :

- soit par le *menu d'aide* dans la barre-menu principale, qui donne accès via un index, à une centaine de rubriques d'aides;
- soit, en cours d'opération, via la touche **F1**. La rubrique d'aide correspondant à l'opération en cours apparaît alors en superposition. Il est éventuellement possible de voyager dans les autres rubriques d'aide avant de revenir à l'étape en cours.

Ces écrans d'aide informent sur l'opération en cours et précisent les touches fonctionnelles et leur effet respectif. Ils ont été conçus de manière à permettre l'utilisation du logiciel en dehors de tout support écrit.

### 2.2.1.4. Recherches pré- et post-coordonnées

*GABI* offre les deux stratégies d'accès aux données, ainsi que toutes leurs combinaisons, selon les besoins de l'utilisateur, ou sa connaissance préalable d'éléments des documents recherchés :



- *recherches pré-coordonnées* (par titre de périodique, maison d'édition, université ou descripteur, ...). L'utilisateur connaît au départ avec exactitude l'élément qu'il recherche et l'introduit directement sous sa forme correcte;
- *recherches post-coordonnées*. L'utilisateur cherche des références touchant à une thématique, sans savoir au départ, à quel endroit de la référence ce thème pourrait apparaître. Il fera alors une recherche par combinaison booléenne de mots-clés.

Il n'a pas été jugé nécessaire d'étendre la recherche pré-coordonnée aux titres des documents et au champ auteur pris dans son intégralité, car ce type de recherche paraît aujourd'hui obsolète.

### 2.2.1.5. Paramétrisations

Les modules de communication peuvent être paramétrés en fonction de préférences personnelles ou de la tâche à réaliser (mode d'affichage, d'impression, ou d'exportation, diverses possibilités de classement, formats d'exportation, personnalisation du titre principal, ...). D'autres procédures acceptent également une paramétrisation pour s'adapter aux souhaits de l'utilisateur (évaluation de la nécessité de réorganisation, installation, statistiques, ...).

Enfin, le système présente deux degrés d'ouverture. L'utilisateur normal n'a accès qu'aux fonctions de consultation et d'exploitation de la base de données. Les opérations de gestion et de maintenance sont protégées par un mot de passe, et réservées au gestionnaire de la base.

## 2.2.2. Niveau profond

### 2.2.2.1. Logique de l'utilisateur

Un soin particulier a été accordé à la transparence du logiciel et à son adaptation aux processus cognitifs spontanés de l'utilisateur. Sans cela, l'analyse dans la seule perspective informatique, des fonctionnalités du système et de leurs relations réciproques, pourrait aboutir à un produit d'une rigueur logique implacable mais pénible à utiliser parce que ne correspondant pas au fonctionnement cognitif des utilisateurs.

Ce travail a porté essentiellement sur :

- *le regroupement des fonctions en familles* présentant un lien conceptuel et leur organisation spatiale sur la barre-menu principale, en fonction du balayage visuel gauche-droite;
- *la modularisation du système* : découpage en unités simples et logiques se succédant en cascades (traitement successif de problèmes particuliers isolés, tout en restant inscrit dans une démarche logique plus globale);

- *la levée des ambiguïtés* par la systématisation de la technique des *choix forcés*;
  - *le centrage sur le produit plutôt que le processus* : par exemple, il n'est pas nécessaire de se préoccuper de l'existence d'un chemin spécifié pour une opération de transfert sur disque. *GABI* analyse la demande et réalise de manière transparente toutes les opérations nécessaires au niveau de l'*operating system*;
  - *le traitement des erreurs* : les opérations erronées sont empêchées et, lors d'une sollicitation inattendue, une confirmation préalable est demandée (communication de la totalité de la base en cours, effacement d'un nombre important de références, ...);
  - *la gestion des touches actives* : standardisation de l'effet de certaines touches et prise en compte de la représentation spatiale du système par le sujet<sup>10</sup>.
- Enfin, dans un certain nombre de cas, des possibilités parallèles permettent un même résultat (*touches de raccourcis*).

#### 2.2.2.2. Vérification automatique des doublons

Afin d'éviter l'introduction involontaire de doublons, *GABI* vérifie en cours de saisie, si une autre référence vraisemblablement identique n'a pas déjà été introduite. Le cas échéant, il en avertit l'utilisateur en temps réel et lui présente la référence similaire, laissant à l'utilisateur le choix de poursuivre ou non l'introduction. L'apparition de la moindre divergence dans la comparaison des premiers éléments des références conduit *GABI* à les considérer comme différentes.

Une vérification similaire s'opère lors de l'importation de références. Le contrôle se fait alors en temps différé. *GABI* crée un fichier des éventuels doublons présents dans le fichier importé. À la fin de la procédure, *GABI* indique le nombre de doublons vraisemblables rencontrés. Il est alors possible de les importer, en mode manuel, à partir du fichier d'identiques créé.

#### 2.2.2.3. Enregistrement retardé

Lors de la saisie d'une référence, l'enregistrement effectif des données sur le disque est différé. Il ne se fait qu'en phase finale, lorsque tous les éléments ont été

---

<sup>10</sup> Le cas de la touche **ESC** est intéressant. Conformément à beaucoup de logiciels, une «signification» d'échappement à la procédure en cours avec retour à l'étape antérieure et abandon des éventuelles modifications lui est attribuée. Ceci permet de renforcer la représentation mentale d'une structure arborescente du logiciel dans laquelle on progresse de la racine vers les ramifications, en ayant à tout moment la possibilité de revenir au branchement précédent. Pourtant, il s'agit là d'une représentation inexacte de la structuration interne de l'application. Il faut d'ailleurs souligner combien les effets obtenus peuvent être en réalité différents, sous leur identité apparente de signification.

introduits et que l'utilisateur a pu vérifier la forme que prendra la référence en normes A.P.A. et éventuellement corriger l'un ou l'autre élément. Ceci permet d'interrompre à tout moment la saisie, sans risque d'altération des fichiers et a également comme avantage d'accélérer la saisie en supprimant les temps de latence intermédiaires d'enregistrement sur disque.

#### 2.2.2.4. Contrôle permanent de l'état du système

Afin d'éviter toute situation susceptible de provoquer un arrêt inopiné de l'application, *GABI* exerce un contrôle permanent de l'état du système. En cas de problème, il en avertit l'utilisateur, lui en spécifie la source et si possible, le moyen de le surmonter. Selon la gravité, il permet ou non la poursuite de l'application.

Dès le démarrage, il vérifie la version du système d'exploitation, la présence des fichiers indispensables à son fonctionnement et l'adéquation de la configuration du système.

Ensuite, de manière permanente, *GABI* vérifie la quantité de mémoire vive disponible. Si celle-ci s'abaisse sous une certaine limite, il empêche le déroulement des opérations lourdes et enjoint l'utilisateur à libérer de la mémoire, par exemple en éliminant des programmes résidents ou en effaçant des étapes de recherches inutiles.

De même, il surveille le bon déroulement de chaque opération d'entrée/sortie (I/O) sur disque ou sur imprimante, analyse les problèmes éventuels, explique la source de l'erreur à l'utilisateur et le guide dans les mesures de correction à effectuer.

Enfin, *GABI* s'adapte sans intervention de l'utilisateur, à l'environnement *hardware*. Ses attributs d'écran se sélectionnent automatiquement en fonction de la carte contrôleur d'écran (monochrome ou couleur) et les appréciations de durée d'opération qu'il propose, sont fonction du processeur.

### 2.3. Gestion de la mémoire vive

Suite aux développements liés aux aspects décrits plus haut, le problème de l'espace mémoire nécessaire pour faire fonctionner l'application s'est rapidement posé<sup>11</sup>. Dans sa version actuelle, en utilisant le mode de compilation le plus économique mais sans segmentation de la mémoire, *GABI* occuperait 635 Ko de

---

<sup>11</sup> L'ensemble des programmes sources constituant *GABI APA Style* version 2.00 dépasse les 24 000 lignes de codes.

mémoire vive, auxquels s'ajouteraient 130 à 250 Ko pour le stockage des variables plus la place occupée par le système. L'ensemble dépassant largement les 640 Ko exploitables sous DOS et *Clipper summer 87* ne permettant pas l'exploitation des extensions de mémoire (voir *supra*, section 1.3.2), un travail important a dû être entrepris pour réduire significativement l'occupation mémoire de l'application.

En définitive, un total de 445 Ko minimum et idéalement 540 Ko suffit au fonctionnement de *GABI*. Pour autant qu'un périphérique de stockage de masse rapide soit utilisé, les délais d'attente sont imperceptibles. Néanmoins, il faut espérer que les versions ultérieures de *Clipper* permettront de gérer les extensions de mémoire, ne fût-ce que pour y stocker les variables.

### 2.3.1. Rationalisation des lignes du code

La première technique utilisée, fut une rationalisation du code source de manière à le dépouiller de tout élément superflu :

- *standardisation* des opérations similaires et *élimination des redondances*;
- développement d'*algorithmes plus concis*. Dans ce cadre, diverses stratégies et structures de programmation ont été évaluées du point de vue de leur performance;
- *intégration* dans les procédures mères de petites procédures de clarification de structure.

### 2.3.2. Structuration en *overlays*

#### 2.3.2.1. Principe

Le principe des *overlays* est de ne placer en mémoire vive, en plus des éléments de base indispensables, que les procédures effectivement utilisées à un moment donné. Chaque fois que nécessaire, le programme opère un échange : les premiers éléments sont éliminés de la mémoire pour faire place aux seconds. Cette technique permet des économies de mémoire appréciables, mais présente deux inconvénients :

- elle nécessite *une structuration et un découpage soignés* du programme en modules indépendants afin d'éviter les conflits d'accès à la mémoire. Pour maximiser les économies, la taille des blocs doit être équilibrée, car la quantité de mémoire réservée pour ces *overlays* est fonction du plus important d'entre eux.
- elle implique des *accès disques fréquents*. Avec un disque dur rapide, le délai occasionné est quasi insensible, mais avec un système à disquette, il peut rapidement devenir problématique. Il est aussi nécessaire d'étudier la structure de l'application de manière à éliminer tout accès disque inutile.

### 2.3.2.2. *Overlays* statiques, *overlays* dynamiques

L'utilisation de la version 5 de *Clipper* aurait offert l'avantage de permettre la création automatique d'*overlays* dynamiques. *GABI* a pourtant été développé selon la technique des *overlays* statiques car, avec une application rigoureusement structurée, les *overlays* dynamiques provoquent un ralentissement de fonctionnement : étant de taille extrêmement réduite (2 Ko environ), le système est obligé de « *switcher* » perpétuellement avec le disque dur.

### 2.3.2.3. Application à *GABI* : trois séries et deux niveaux

Une première application des *overlays* à *GABI* concerne les procédures liées aux écrans d'aide qui consomment près de 9 000 lignes de programmation. Ceci a permis de réduire la place mémoire qui leur est réservée, de 168 Ko à 11 Ko seulement. Afin de ne pas multiplier exagérément le nombre d'*overlays*, les 100 programmes d'aide ont été regroupés en 25 *overlays*.

Des *overlays* ont également été constitués pour les grandes fonctions du logiciel (les modules 1, 2, 3, ..., Y du schéma présenté plus bas). Ces fonctions correspondent globalement aux grandes familles d'opérations de la barre-menu principale (recherche, gestion, communication, outils, ...). Cependant, l'existence de disparités importantes en taille de code entre ces différents groupes<sup>12</sup> a amené à un découpage complémentaire de certains groupes de fonctions.

De même, bien que structurellement participant à un autre niveau, un *overlay* d'initialisation des variables et de définition de l'environnement a été ajouté à cette série. Il reprend également la phase d'accueil et d'ouverture de l'application. En définitive, cette série comporte 9 blocs d'*overlays*. Pour rendre possible l'accès, à partir de n'importe quel module, à n'importe quelle rubrique d'aide, il était nécessaire de séparer les deux séries d'*overlays* pour permettre des « glissements » parallèles et indépendants.

En dernière analyse, il subsistait un module de taille beaucoup plus importante et qui ne pouvait être éclaté en unités séparées, étant donné leur large interdépendance : le module Gestion. Une troisième série de 18 *overlays* a dès lors été conçue. Celle-ci se situe à un deuxième niveau : elle est emboîtée dans l'*overlay* GESTION, ce qui permet de ramener le code de ce module de 4 000 à moins de 2 000 lignes plus une zone mobile de 270 lignes. Ces *overlays* apparaissent sous l'appellation « Sous-Mod » dans le schéma ci-contre.

La mise en place de ces 52 *overlays* en 3 séries réparties en 2 niveaux a permis de ramener la place mémoire nécessaire au chargement du programme de 635 Ko à 321 Ko (gain : 49,45 %).

<sup>12</sup> Le module OUTILS complet par exemple, prend plus de 4 300 lignes de code, alors que le module COMMUNICATION n'en occupe que 1 700.

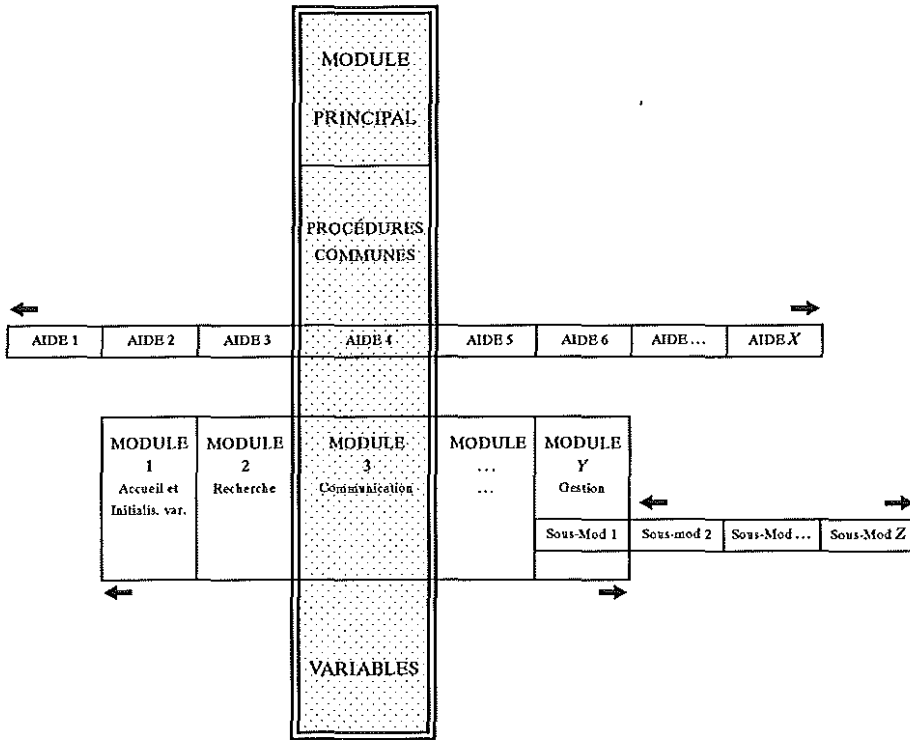


Fig. 2.- Représentation graphique de la structure des *overlays* dans *GABI APA Style* : trois séries en deux niveaux.

La représentation graphique de la structure des *overlays* a été simplifiée pour en faciliter la lecture. Le grisé représente la place réellement occupée en mémoire vive avec les « niches » dans lesquelles « coulisent » les *overlays* des différentes séries.

### 2.3.3. Sauvegardes temporaires

Parmi les variables utilisées, deux catégories sont particulièrement gourmandes en mémoire : celles permettant le stockage d'images d'écrans qui accélèrent sensiblement les nettoyages d'écran, mais coûtent 4 Ko par écran sauvé, et celles contenant les résultats de recherches à nombre élevé d'occurrences (malgré un codage économique, le résultat d'une recherche ayant aboutit à plusieurs centaines d'occurrences, « coûte » plusieurs Ko.)

Pour éviter l'encombrement de la mémoire suite à la multiplication de macro-recherches par exemple, le principe de la sauvegarde temporaire leur a été appliqué : leur contenu est stocké sur disque et n'est rappelé qu'en cas de nécessité.

Grâce également à l'utilisation d'autres procédés techniques, 200 à 230 Ko de mémoire disponibles pour les variables suffisent pour permettre l'utilisation intensive de l'application.

## 2.4. Ouverture du logiciel

*GABI* permet deux types d'interfaçage. La première est orientée vers l'utilisation de la référence en tant que texte et la seconde, en tant qu'enregistrement de base de données.

### 2.4.1. Exportation en format texte

Des références sélectionnées par une recherche peuvent être exportées dans leur forme A.P.A. vers un fichier DOS, pour être récupérées par un traitement de texte par exemple. Plusieurs formats sont proposés :

- *format ASCII simple* : les références sont stockées en caractères ASCII, par conséquent sans mise en forme. Le document peut alors être récupéré tel quel par la plupart des éditeurs ou traitements de texte. L'absence de formatage ne permet cependant pas de distinguer la partie devant être en italique ou soulignée;
- *format ASCII marqué* : des repères sont ajoutés en début et fin des zones à mettre en italique. Il suffit alors de récupérer le document sous un traitement de texte et de créer une macro-commande qui recherche ces repères et applique automatiquement le format de caractère adéquat. Un exemple d'une telle macro est fourni pour le traitement de texte *WORD 5* de *Microsoft*<sup>®</sup>;
- *formats particuliers* : il est possible de créer des interfaces de traduction qui ajoutent aux données les codes de contrôle nécessaires pour permettre leur récupération par des traitements de texte spécifiques. Cela nécessite cependant une connaissance approfondie de ces codes et des formats spécifiques des traitements de texte.

### 2.4.2. Format d'échange ISO 2709

*GABI* permet également d'exporter et d'importer des références en conservant leur statut de données stockées dans des champs différenciés<sup>13</sup> via le

---

<sup>13</sup> L'exportation en format texte expliqué plus haut est un produit achevé. Il n'est plus possible de manipuler automatiquement les éléments constitutifs de la référence : auteurs, maison d'édition, date d'édition, titre du périodique, ...

format d'échange ISO 2709, défini par l'*International Organization for Standardization* comme standard de communication de données bibliographiques entre machines. Cette option présente plusieurs avantages :

- permettre la récupération de notices créées sur d'autres machines utilisant *GABI*, ce qui ouvre la voie à la saisie décentralisée;
- offrir une technique rapide et économique de « *backup* » de la base de données (en format ISO, une référence occupe en moyenne 0,5 Ko au lieu des 4 à 5 Ko nécessaires index compris, dans sa forme exploitable en base de données);
- ouvrir la porte à l'échange de données avec d'autres systèmes de gestion de bases de données, Beaucoup d'entre eux (*STAIRS*® ou *LIBER*™, ...) possèdent un module de communication selon le standard ISO 2709. Il suffit de définir le protocole de communication en spécifiant la signification des étiquettes (*tag*) des composants de la référence, et le fichier créé par *GABI* devient exploitable par l'autre logiciel.

À titre d'exemple, voici un enregistrement d'une référence en format ISO créé par *GABI*.

```
00524N 0100196 34 00101000000020140010005002002400700200
26010004002802003500320500170067051026008405202001101000170130
15004601471700060193200019019930003602184000280254450037028290
00090319▲FAPSE 118▲010791/000000▲3▲F▲V▲AN▲▼Service de Psych
ologie du Langage▲▼HANDICAP MENTAL▲▼LANGAGE -- DEVELOPPEMENT
▲▼LANGAGE -- TROUBLE▲▼Rondal, J.A.▼1▲▼Language development
and mental retardation▼▲▼1987▲▼Oxford▼Angleterre▲▼Blackw
ell, Philadelphia Lippincott▲▼W. Yule & M. Rutter (Eds.)▲▼La
nguage development and disorders▼▲▼248--261▲↵
```

ce qui se traduit par

Rondal, J. A. (1987). Language development and mental retardation.  
In W. Yule & M. Rutter (Eds.), *Language development and disorders*  
(pp. 248-261). Oxford, Angleterre: Blackwell, Philadelphia  
Lippincott.

NUM REF : FAPSE 118                      CREE LE : 01/07/91                      LANGUE : Anglais  
Service de Psychologie du Langage

HANDICAP MENTAL                      LANGAGE - DEVELOPPEMENT                      LANGAGE - TROUBLE



### 3. Phases de développement du logiciel

#### 3.1. Version 1.3

La première version de *GABI* date de 1991. Les objectifs et la philosophie de l'application étaient déjà définis et son analyse générale, largement développée, mais cette version initiale n'offrait que les fonctions primitives du logiciel (*ajout de références, visualisation, un embryon de recherche et quelques outils d'impression*). Sa seule ambition était de permettre un test grandeur nature de la structure des fichiers et index et du mode d'approche de la référence.

#### 3.2. Version 2.00

La version actuelle a été développée à partir d'octobre 1991 sur base des constatations relevées lors de la phase précédente. Elle s'en distingue principalement par :

- des modifications dans la structure des fichiers et index;
- l'amélioration du module de recherche, en particulier dans ses possibilités d'utilisation de troncatures et d'opérateurs booléens;
- la diversification des possibilités d'impression;
- l'incorporation des modules de modification et de suppression de références, d'importation et exportation sous divers formats;
- l'ajout d'utilitaires;
- l'introduction du système d'aide;
- enfin, tout le travail périphérique de gestion de la mémoire, d'ergonomie logicielle, de standardisation des processus, de gestion des erreurs et d'aide à la saisie a été réalisé.

#### 3.3. $\beta$ Test

La dernière étape a consisté, d'une part en une optimisation et d'autre part, en un toilettage final à travers une phase de  $\beta$  test.

Un échantillon de sujets, différenciés quant à leur compétence informatique, a été confronté à *GABI APA Style*. Leurs réactions, stratégies et remarques ont été étudiées afin d'observer leur compréhension des opérations, de conforter ou infirmer les hypothèses quant à la logique de l'utilisateur, d'améliorer des procédures (par exemple l'installation d'une technique de « *trace* » pour les combinaisons booléennes), de mettre en évidence les imprécisions et de rencontrer des conditions d'erreur non prévues.

Si ce  $\beta$  test a fourni des renseignements précieux, son ampleur limitée ne permet pas d'espérer la levée de tous les « bugs » ou défauts dans la version actuelle. Le dernier et véritable filtre d'une application informatique est sans conteste sa diffusion réelle auprès des utilisateurs finaux.

## 4. Principales fonctionnalités<sup>14</sup>

### 4.1. Recherche

Des recherches sélectives peuvent s'opérer au moyen de 15 critères. Certains portent sur l'ensemble des documents et concernent soit les éléments bibliographiques de la référence (nom d'auteur, année de publication, ...), soit les informations complémentaires (descripteurs, langue, service ou sein duquel le document a été réalisé, numéro d'identification, ...), d'autres sont spécifiques à certains types de documents (titre de périodique, maison d'édition, ...). Enfin, il est possible de faire des recherches par mots-clés (n'importe quel mot significatif de la référence ou des descripteurs).

La *troncature droite* est permise pour tous les critères non soumis au choix forcé (auteur, mot-clé, descripteur, titre de périodique, maison d'édition, université de défense d'une thèse, ...). Elle n'est pas limitée quant au nombre de caractères de la racine.

Toutes les *combinaisons booléennes* multi-critères sont possibles. Pour éviter les confusions, elles doivent être mono-nomiques à l'intérieur d'une séquence de recherche. Pour réaliser des recherches complexes mêlant différents opérateurs, il suffit de décomposer l'équation en structures élémentaires ne comportant qu'un seul type d'opérateur, puis de combiner avec les autres opérateurs les résultats obtenus. Cette démarche par étape est plus accessible à l'utilisateur et moins sujette aux erreurs de saisie.

Par exemple, pour trouver le résultat de

$$(A \text{ OR } B \text{ OR } C) \text{ AND } (D \text{ OR } (E \text{ NOT } F)),$$

on fera successivement :

étape 1 :  $A \text{ OR } B \text{ OR } C$

étape 2 :  $E \text{ NOT } F$

étape 3 :  $D \text{ OR } \text{étape 2}$

étape 4 :  $\text{étape 1 AND } \text{étape 3}$

<sup>14</sup> Pour une description détaillée, le lecteur se référera au mode d'emploi de *GABI APA Style*.

## 4.2. Gestion

Cet ensemble de fonctions permet le développement et la gestion de la base de données. Pour des raisons évidentes, ce module n'est accessible qu'au gestionnaire de la base de données et est protégé par *mot de passe*. Il est possible d'*ajouter* des références, d'en *modifier*, d'en *supprimer* ou d'en *importer* via le format ISO.

L'ajout de nouvelles références ainsi que leur modification sont séquentiels. L'utilisateur est guidé et assisté par le logiciel de manière à simplifier la saisie et limiter les possibilités d'erreur.

Dans les deux procédures d'incorporations de nouvelles références (ajout et importation), un système de *vérification automatique* permet de repérer les doublons éventuels (voir section 2.2.2.2). Dans le cas de l'importation, ce contrôle peut être déconnecté.

## 4.3. Communication

Les références sélectionnées, voire l'ensemble des références de la base, peuvent être communiquées de trois manières : *l'affichage* à l'écran, *l'impression* ou le *stockage* sur disque. Chacun de ces modes de communication peut être paramétré par l'utilisateur.

La séquence d'affichage à l'écran n'est pas linéaire. Il est possible de naviguer à sa guise dans le sous-ensemble de références sélectionné.

Outre la possibilité d'exportation en format texte, *GABI* permet l'exportation de références en format ISO permettant l'échange de données entre machines (voir section 2.4.2).

## 4.4. Outils

Cette rubrique compte huit utilitaires :

- le *module préférences* permet la paramétrisation des options de communication du logiciel. Il permet aussi de créer au moyen des séquences d'échappement, des pilotes d'imprimantes originaux, ainsi que des utilitaires de conversion en formats spécifiques de certains traitements de texte;
- le *module fichiers d'autorité*. Étant donné les conséquences d'une modification malheureuse à ce niveau, ce module est réservé au gestionnaire de la base et protégé par mot de passe. En fonction du fichier d'autorité, diverses opérations peuvent être réalisées : corrections orthographiques, condensation

(transfert des notices d'un élément vers un autre), ajout ou suppression d'éléments;

- le *module d'impression d'index* permet l'impression du contenu des fichiers d'autorité avec le nombre d'occurrences de chaque élément et, au choix, la liste des numéros de référence auxquels ils sont liés, afin de permettre par exemple la constitution de catalogues imprimés;
- le *module thesaurus*, non accessible dans la version actuelle, permettra de gérer les relations entre descripteurs (voir perspectives d'avenir, section 6);
- le *module statistiques* permet d'obtenir des informations sur le contenu actuel de la base de données (nombre et proportion de documents par catégories, par date de publication, par langue, ..., proximité des auteurs, fréquence de publication par périodique, par éditeur, ..., occurrences d'utilisation de mots, de descripteurs, ...);
- le *module réorganisation* permet la maintenance régulière des fichiers pour récupérer de l'espace disque et accélérer le travail du logiciel. Une évaluation préalable de la nécessité de réorganisation est proposée;
- le *module d'installation*. Selon le type d'unité de stockage choisi, une installation « complète » totalement opérationnelle du système; est proposée, ou une version réduite, sur disquette, destinée à servir de « vecteur de transport » du logiciel, en vue de son installation sur un autre ordinateur;
- le *module d'effacement des étapes de recherche préalables* permet d'éliminer les étapes de recherche antérieures et récupérer ainsi de la mémoire.

#### 4.5. Aide

Dans le module d'aide, deux possibilités sont offertes :

- un classique « *À propos de ...* » indique la version utilisée, la quantité de mémoire vive disponible et la date de la dernière modification des fichiers;
- l'accès à travers un index à une centaine de *rubriques d'aides*. Cette aide est également accessible à tout moment via l'emploi de la touche **F1** (voir section 2.2.1.3).

## 5. Données techniques

### 5.1. Configuration nécessaire

- PC compatible; idéalement, processeur 80286 ou plus puissant.
- Système d'exploitation égal ou supérieur à MS-DOS 3.3™.
- Mémoire vive disponible pour l'application d'au minimum 445 Ko; idéalement, égale ou supérieure à 550 Ko.
- Disque dur rapide.
- Unité de disquette 720 Ko ou plus pour l'installation.
- N'importe quel type d'écran graphique ou texte.
- Toute imprimante, pour autant que l'on dispose du pilote (*driver*) correspondant ou qu'on le crée.
- Pour permettre l'ouverture d'un nombre suffisant de fichiers, le fichier CONFIG.SYS doit comporter la commande `files = 85` et le fichier AUTOEXEC.BAT, la commande `set clipper = f80`. GABI peut adapter lui-même ces éléments au moment de l'installation.
- L'utilisation de la commande FASTOPEN munie d'un paramètre supérieur ou égal à 80 est optionnelle mais accélère sensiblement les accès disques répétés. Il en va de même de tous procédés d'accélération des accès disques (mémoire cache, ...).

### 5.2. Tests de performance

Quelques tests de performance ont été réalisés sur une base de 1 800 références, avec un PC 386 cadencé à 25 Mhz et muni d'un disque dur de temps moyen d'accès de 19 ms. Ils permettent la comparaison éventuelle avec d'autres applications.

Les mesures de vitesse portent sur des situations extrêmes permettant d'apprécier les limites du système. En usage courant, les délais sont sensiblement plus courts, voire imperceptibles.

#### 5.2.1. Recherche

- ◇ Temps nécessaire pour la recherche par fichiers d'autorité la plus lourde possible (recherche par mots-clés avec troncature illimitée et racine vide, ce qui équivaut à un OU logique avec 2 570 mots-clés) : 3' 05"
- ◇ Recherche équivalente sur le champ AUTEUR (258 auteurs différents) : 45"
- ◇ Temps de repérage de 1 533 références sur 1 800 sans utilisation de fichiers d'autorité (LANGUE = FRANÇAIS) : 21"

### 5.2.2. Ajout

- ◊ Délai de vérification de redondance pour une référence avec un seul auteur apparaissant dans plus de 350 autres références : 4"
- ◊ Espace disque total moyen par référence (y compris le dictionnaire) : < 5 Ko

### 5.2.3. Importation

La vitesse d'importation de références en mode ISO est liée au nombre de références présentes dans la base de données. Au-delà de 500 références, on observe une quasi stabilisation de la vitesse autour d'une moyenne horaire de 1 000 références/heure, soit 17 réf./minute.

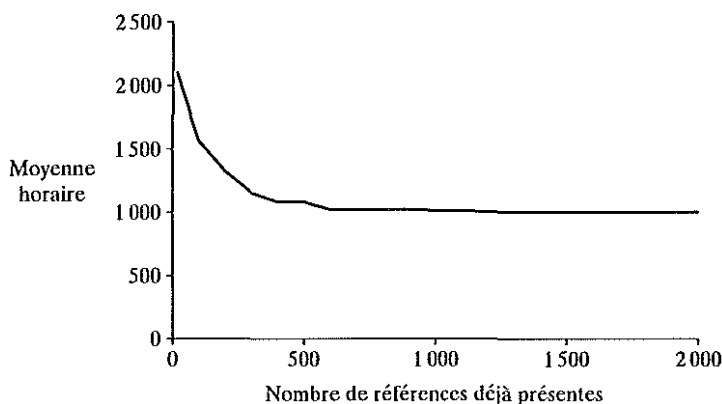


Fig. 3.- Vitesse d'importation de références en fonction de la taille de la base de données

Ces valeurs correspondent au cas d'importation d'un nombre limité de références (100 à 200) dans une base fraîchement réorganisée. Elles peuvent varier sensiblement dans les autres cas. Ainsi, lors de l'introduction de fichiers de plusieurs centaines de références, la moyenne peut s'affaïsser jusqu'à 600 références/heure pour les dernières références (40 minutes sont nécessaires pour importer un fichier ISO de 600 références, soit une moyenne générale de 900 références/heure). Ceci s'explique par l'absence de réorganisation intermédiaire des fichiers.

### 5.2.4. Exportation

- ◊ Vitesse d'exportation en mode ISO, sur disque dur : 12 600 réf./heure
- ◊ Taille d'un enregistrement ISO : 0,5 Ko
- ◊ Vitesse d'exportation en mode ASCII, sur disque dur : 15 000 réf./heure
- ◊ Taille d'un enregistrement ASCII complet : 0,44 Ko

### 5.2.5. Calcul statistique

- ◊ Durée moyenne : 1 seconde pour 50 références

### 5.2.6. Réorganisation<sup>15</sup>

- ◇ Temps nécessaire pour 1 800 références : 6' 15"

### 5.2.7. Installation

- ◇ Installation complète sur disque dur : 29'
- ◇ Installation d'une version de transport sur disquette (fonction de la rapidité d'accès de l'unité de disquette) : ± 1' 20"

## 6. Perspectives d'avenir

Certaines pistes d'amélioration du logiciel évoquées ci-après concernent la structure profonde de l'application, d'autres sont des améliorations de surface. Aucune n'a été soumise à une analyse détaillée quant à la faisabilité et l'intérêt réel. S'y ajouteront les remarques et propositions des utilisateurs finaux.

### 6.1. Élargissement à d'autres normes

Permettre le formatage des références selon plusieurs normes au choix serait séduisant. Cette idée devra cependant être étudiée du point de vue du rapport coût-bénéfice pour l'utilisateur car elle risque d'impliquer un alourdissement considérable des procédures de saisie de références par exemple. Une solution alternative pourrait être de développer des versions parallèles de *GABI* (ISBD Style, Style Presses Universitaires de France, ...).

### 6.2. Ajout d'un résumé

L'ajout d'un résumé à la référence pourrait s'avérer intéressant, pour autant que l'accroissement du fichier dictionnaire, de ses fichiers relationnels et index qui en résulterait soit tolérable.

---

<sup>15</sup> Il s'agit de l'opération la plus lourde du logiciel (17 bases de données et 45 index).

### 6.3. Thesaurus en ligne

Un module complet de gestion de thesaurus permettrait d'éviter la redondance de l'utilisation d'un outil supplémentaire, mais surtout d'exploiter le réseau de relations arborescentes du thesaurus (extension de la recherche aux termes spécifiques, saisie et recherche assistée par guidage dans le réseau des relations associatives (proposition automatique de termes sémantiquement proches, renvoi transparent des termes exclus vers les termes acceptés, ...).

Ce module devrait être développé sous une forme double : l'une incluse dans *GABI* et l'autre fonctionnant sur les mêmes fichiers de descripteurs, mais indépendamment de *GABI*, ce qui permettrait l'exploitation d'un thesaurus unique pour diverses applications extérieures.

### 6.4. Modifications non séquentielle

Un mode pleine page pourrait s'avérer intéressant pour la modification de références.

### 6.5. Recherche différée

La stratégie de recherche de *GABI* est le repérage effectif de tous les documents correspondants à la question, ce qui consomme du temps, lorsque le nombre d'occurrences est élevé. Or, ce repérage n'est réellement indispensable que lors de la communication de références (affichage, impression ou exportation) et de la combinaison de plusieurs étapes. Dans les autres cas, l'information sur le nombre d'occurrences est suffisante.

Les fréquences d'occurrences étant immédiatement accessibles sous *GABI*, il serait envisageable de ne réaliser réellement le repérage que dans les deux cas mentionnés. Ces recherches seraient alors ralenties, mais les autres en seraient accélérées et la mémoire vive, économisée d'autant. Cependant, Il faudrait vérifier si la combinaison booléenne d'étapes de recherche à nombre élevé d'occurrences ne provoquerait pas alors des délais inacceptables (concentration de procédures longues).

### 6.6. Suppression du rappel intermédiaire

Lors de la récupération d'une équation préalable (recherche par étape), celle-ci est rappelée en une étape intermédiaire. L'élimination de cette redondance améliorerait la lisibilité du protocole de recherche et allégerait d'autant la charge en mémoire vive. Cette évolution implique cependant des modifications importantes dans les algorithmes de recherche.



### 6.7. Opérateurs de contexte

Dans la recherche par mots-clés, il serait intéressant d'offrir en plus des opérateurs booléens, d'autres possibilités de combinaisons comme la recherche par cooccurrences, ou même avec des opérateurs de proximité<sup>15</sup>.

### 6.8. Utilisation de la souris

Étant donné sa commodité et son caractère de plus en plus commun, *GABI* devra permettre dans l'avenir l'utilisation de ce périphérique pour la navigation dans les menus.

### 6.9. Adaptation au *Macintosh*

L'importance de la diffusion des *Macintosh* impose d'envisager une adaptation *Macintosh* de l'application. Pour éviter la réécriture de l'ensemble du programme dans un langage disponible dans cet environnement, il faudra probablement transiter par un utilitaire permettant d'ouvrir une fenêtre DOS sur *Macintosh* et d'y faire tourner une application écrite pour *PC*. Le degré de compatibilité devra être évalué et les conséquences sur la performance de cette couche supplémentaire, testées.

### 6.10. Développement de pilotes particuliers

Enfin, une perspective d'avenir est le développement de modules de conversion vers des formats de traitements de texte spécifiques (*WORD 5*, *WordPerfect*®, *WORD Windows*®, ...). La potentialité existe, mais une application concrète n'a pas encore été réalisée. Il est probable que des problèmes particuliers ne manqueront pas de surgir à cette occasion.

## Bibliographie

American Psychological Association (1984). *Publication manual of the American Psychological Association* (Third Edition). Washington, WA : Author.

BEUTLER E., SCHNEIDER K., KENNEY N. & PELINA E. (1990). *Reference manager* [Computer program]. Dusseldorf, Allemagne : Research Information Systems.

---

<sup>15</sup> Les opérateurs de cooccurrence permettent de rechercher deux termes présents simultanément dans la même phrase, ou dans le même champ. Les opérateurs de proximité sont encore plus précis. Ils permettent de retrouver les références dans lesquelles un premier terme se trouve à un nombre de mots spécifié d'un second. L'ordre dans lequel ces deux termes doivent apparaître peut également être précisé.

- BONGARTZ L. (1989). *Clipper Tools One* [Computer program]. Los Angeles, CA : Nantucket Corporation.
- COIBION C. (s.d.). *G.T.I. : Gestion de Thesaurus Interactive* [Computer program]. Liège, Belgique : Université de Liège.
- ELWORK A., HALL P., STONE B., & YOKELSON D. (1988). *Manuscript manager APA Style* [Computer program]. Washington, DC : Pergamon Press.
- International Organization For Standardization (1981). *Norme ISO 2709 : Documentation — Disposition des données sur bande magnétique pour l'échange d'informations bibliographiques*. Genève, Suisse : Author.
- LESENFANTS M. (1991). *Publications des membres académiques de la Faculté de Psychologie et des Sciences de l'Education : Création d'une base de données selon les normes de l'A.P.A.* Travail de fin d'études en bibliothéconomie non publié, Institut supérieur d'enseignement des Sciences de l'Information et des Sciences sociales de la province de Liège. Liège, Belgique.
- RUSSEL B. (1987). *Clipper* [Computer program]. Los Angeles, CA : Nantucket Corporation.
- STRALEY S.J. (1987). *Programming in Clipper : The definitive guide to the Clipper dBase compiler* [Second edition]. Reading, MA : Addison-Wesley Publishing Company.
- VANDERHEYDEN R. (1990). *Catalogue analytique des mémoires de psychologie de l'Université de Liège de 1985 à 1989 suivi d'un thesaurus*. Travail de fin d'études en bibliothéconomie non publié, Institut supérieur d'enseignement des Sciences de l'Information et des Sciences sociales de la province de Liège. Liège, Belgique.