

Université de Liège

Faculté de Psychologie et des Sciences de l'Éducation

G A B i A P A Style

**Un Gestionnaire Automatisé de Bibliographies
selon les normes de l'A.P.A.**

(American Psychological Association)

**Année académique 1991-1992
Maîtrise en Bibliothéconomie**

Paul THIRION

Le développement de GABI APA Style est une longue histoire ...

Si les développeurs étaient conscients au départ des multiples difficultés de détails qu'ils rencontreraient inévitablement au cours de la création d'un logiciel, sans doute, bien peu verraient le jour ! On considère d'ailleurs que pour avoir une idée réaliste du temps qui sera nécessaire à l'écriture d'un logiciel, il faut multiplier au moins par deux et demi le temps estimé au départ ! Dans le cas de GABI, cette proportion est sans doute plus déséquilibrée encore ...

Mais on se prend au jeu, on se passionne et finalement ... on développe un syndrome pré-œdipien de relation quasi fusionnelle avec la machine ! On ne voit plus que clés d'index, mémoire, gestion d'erreurs, procédures, "bugs", ...

Aussi, je voudrais avant tout dire merci à Dophie et Valentin, qui ont tant de fois accepté de "laisser papa travailler" avec son drôle de jouet, si attirant ... alors qu'il aurait été tellement gai d'aller se promener tous ensemble, ou de pouvoir frapper aussi sur tous ces petits boutons, comme papa ...

Merci surtout à Veronique, qui non seulement a accepté de me décharger de quantités de tâches journalières, mais aussi a si souvent toléré ces "Je viens dans deux minutes..." qui se transformaient en réalité en larges morceaux de nuits ... et a supporté patiemment et sans jalousie, pendant près de deux ans, l'intrusion de cette GABI rivale qui m'occupait parfois tant l'esprit ... Sans toi, rien n'aurait été possible.

Merci à Myriam qui a pris en charge, dans le cadre de son travail de fin d'études, la saisie et l'indexation des documents de la base UDPSY.

Merci aussi aux personnes qui de près ou de loin, ont contribué à la réalisation de GABI APA Style : les unes en acceptant de tester la version de β -test, les autres en relisant des parties de l'anti-dictionnaire.

Merci aussi à Madame De Keyser, Messieurs Denooz et Crahay, qui ont accepté de s'astreindre à la lecture de ces pages si peu poétiques.

Enfin, merci à mon PC qui ... jusqu'à présent ... ne m'a pas encore fait faux bond, même s'il m'a donné nombre de sueurs froides !

Paul Thirion,
Hévermont, le 18 mai 1992

AVERTISSEMENT

GABI APA Style doit être considéré comme ce qu'il est, c'est à dire une application documentaire informatique développée dans un contexte spécifique, pour répondre à des besoins particuliers, et qui n'a aucune ambition ni d'universalité, ni même de concurrencer des applications professionnelles.

Il est le fruit du travail d'un individu isolé, non pas informaticien ou analyste programmeur, ni même véritable bibliothécaire, mais d'un simple psychologue, clinicien de formation, qui s'est retrouvé un jour responsable d'une Unité de Documentation et a pris goût à la micro-informatique, puis s'est lancé, en autodidacte, dans la programmation.

Il est donc possible que certains choix de programmation soient discutables pour des spécialistes, voire même que l'analyse générale de l'application eût pu être optimisée. D'ailleurs, mes techniques de programmation et d'analyse ont subi une profonde maturation au cours des presque 2 ans de gestation de l'application. Un certain nombre de procédures ont même été complètement réécrites. Il est également probable que malgré une phase de β -test, un certain nombre de "bugs" subsistent dans la version actuelle.

Quoi qu'il en soit, ce qui a toujours guidé les décisions, c'est le souci de pragmatisme et de convivialité.

- **Pragmatisme**, car à tout moment l'objectif recherché était l'efficacité réelle : que le programme donne les résultats escomptés, en occasionnant des délais minimum, même si pour y arriver, il fallait évaluer les diverses stratégies possibles empiriquement, par essais et erreurs.
- **Convivialité**, car je voulais que tout se passe de manière transparente pour l'utilisateur, que GABI suive la logique de celui-ci et non une logique d'informaticien, et qu'il propose à tout moment des choix simples et évidents, en apparence du moins. Le respect de cette philosophie impliquait bien entendu, l'impossibilité de faire l'économie de la gestion des erreurs.

Avec ses limites, j'espère simplement que GABI pourra atteindre les objectifs qui lui ont été fixés, à savoir, permettre de constituer la mémoire du "*background intellectuel*" de la FAPSE. Si en plus, les membres de la Faculté le considèrent comme un réel outil de travail utile et pratique et non pas comme un jouet mal achevé, et s'ils l'emploient avec un certain plaisir, alors les centaines d'heures d'analyse et de programmation n'auront pas été vaines.

Je reste bien entendu ouvert à toute remarque, critique ou idée d'amélioration ou d'évolution de GABI, car un logiciel doit rester un objet vivant et évolutif, sous peine d'être rapidement abandonné.

Enfin, je tiens d'ores et déjà à m'excuser auprès du lecteur pour les inévitables données techniques qui émaillent le texte. Ne pas les mentionner aurait sans conteste allégé le texte, mais l'aurait aussi considérablement vidé de sa substance, car là se trouvent les fondements mêmes de GABI APA Style et de son développement.

Si d'aventure ces éléments excédaient le lecteur, je le convie à abandonner un instant le texte et à faire quelques essais pratiques sur le logiciel, car la finalité de ce travail ne se trouve pas dans l'exercice de style, mais dans le développement informatique d'un outil que j'espère utile et agréable.

I. INTRODUCTION : MOTIVATION DU DÉVELOPPEMENT DE GABI APA STYLE

A. UN CONSTAT : LA SITUATION DES PUBLICATIONS DES MEMBRES DE LA FAPSE

La décision de mettre sur pied une base de données des publications des membres de la Faculté de Psychologie et des Sciences de l'Education est née de la rencontre de deux démarches différentes sur un même constat.

1. SOUHAIT DU CONSEIL DE FACULTÉ

En 1985, l'Institut de Psychologie et des Sciences de l'Education s'est engagé dans le processus de transformation en Faculté. A cette occasion, un inventaire des travaux et publications de ses membres dut être dressé pour démontrer son dynamisme scientifique.

Il fallut alors constater l'état de désorganisation de ce type d'information. En effet, si, la production scientifique était extrêmement riche, variée et d'un haut niveau de qualité, elle était tout autant "volatile", étant donné l'absence de politique systématique de conservation de traces de ces publications. Par défaut, chacun, dans le meilleur des cas, chaque service, était seul responsable de cette conservation. Cette politique, ou plutôt cette absence de politique eut une double conséquence :

- les publications des chercheurs et même des académiques ayant quitté l'Institut n'étaient conservées que de manière aléatoire. Pour la plupart, elles semblaient très vite dans l'oubli;
- même en ce qui concerne les membres effectifs de l'Institut, la tendance était de ne conserver trace que des publications les plus récentes ou les plus importantes (monographies). Souvent même, seule la référence du document était conservée dans le curriculum, alors que le document lui-même avait disparu et était devenu introuvable.

A ce caractère éphémère et dispersé de notre patrimoine intellectuel, s'ajoutait la piètre qualité des références bibliographiques existantes. A relire les listes collationnées à

l'époque, on constate que la plupart étaient très lacunaires et ne respectaient aucune norme. En fait, 2 ou 3 académiques seulement possédaient une référothèque personnelle un tant soit peu complète, organisée et respectant des normes de présentation largement reconnues.

Aux prises avec cet état de réalité, la constitution de l'inventaire des publications des membres de l'Institut fut une entreprise extrêmement difficile et hasardeuse, dont les résultats ne purent qu'être largement lacunaires et, partant, insuffisamment représentatifs.

Ce constat a eu le mérite d'amener en 1990 le Conseil de ce qui était devenu Faculté, à vouloir arrêter l'hémorragie et se prononcer pour la constitution d'une base de données facultaire de ses productions scientifiques. Il chargea l'Unité de Documentation d'en assurer la réalisation, sans pour autant lui en définir les moyens.

2. L'UNITÉ DE DOCUMENTATION

Parallèlement, l'Unité de Documentation, en pleine réorganisation, se trouvait confrontée à une autre facette du même problème.

Il lui devenait possible via les interrogations de bases de données ON Line et sur Compact Disc, de fournir à ses utilisateurs des références bibliographiques pertinentes de travaux publiés à quelque endroit de la planète, mais par contre, elle n'avait bien souvent pas connaissance des publications des membres anciens ou même actuels de la Faculté. Elle possédait certes en rayons un certain nombre de monographies des membres les plus connus de la Faculté, mais sans pouvoir se targuer, et de loin, d'exhaustivité. La situation était plus grave encore pour les articles de périodiques et surtout pour les rapports de recherche et communications à des congrès ou colloques. Pour ces types de documents, même lorsque l'auteur n'avait pas disparu de la circulation, il n'en possédait souvent lui-même plus aucun exemplaire. Rien ne permettait d'espérer une amélioration structurelle.

Cette situation très inconfortable par rapport aux utilisateurs habituels de l'U.D. était devenue intolérable lorsque, ainsi que cela se produisit à plusieurs reprises, sont arrivées à l'U.D. des demandes de copies de document publiés au sein de la Faculté dont nous n'avions même pas connaissance et dont il était souvent très difficile de retrouver la trace, même auprès des services concernés.

C'est chaque fois l'image de marque de la Faculté et à travers elle, celle de l'Université qui s'en trouvaient ternies.

B. GABI ET LA CONSTITUTION D'UNE BASE DE DONNÉES DES PUBLICATIONS DE LA FAPSE

Face à ce constat initial, il fallait s'engager dans une double démarche :

- En première priorité : élaborer une base de données de ces références pour conserver au moins une trace de leur existence;
- Parallèlement, constituer au niveau facultaire, un fonds des documents primaires correspondants, en ayant pour objectif de conserver au moins un exemplaire de tout document publié par un membre de la Faculté.

Par rapport au premier axe, il eut été possible, beaucoup moins contraignant et assurément plus rapide, de se contenter de la création d'une simple liste reprenant chaque référence exactement telle que son auteur nous l'aurait transmise.

Cependant, s'en tenir à cette stratégie ne semblait guère opportun pour deux catégories de raisons :

- cela eut été privilégier l'aspect **conservatif** en se coupant de toute possibilité d'**exploitation** du fichier. En effet, la recherche critériée de références serait devenue rapidement fastidieuse dès que le nombre total de celles-ci aurait dépassé la centaine.
- de plus, les possibilités d'utilisation effective de ces références s'en seraient trouvées extrêmement limitées, par l'absence de standardisation de leur forme.

Par conséquent, il fallait envisager la mise en place d'un système permettant ¹ :

- de conserver une trace de toutes les références publiées par des membres de la Faculté;
- d'introduire facilement et de manière conviviale de nouvelles références, d'en supprimer et d'en modifier chaque fois que nécessaire;
- de réaliser des recherches sur de nombreux critères, même par des non spécialistes du système;
- de produire des résultats largement et directement exploitables par les utilisateurs de l'U.D. mais aussi par les chercheurs (pour cela, il fallait dès lors utiliser un format de présentation des références bibliographiques clair et largement reconnu dans le monde scientifique concerné).

¹ Remarquons qu'on retrouve là les caractéristiques principales d'un S.G.B.D. (Système de Gestion de Base de Données), avec en plus l'accent mis sur le format de présentation des références.

- d'échanger aisément des données à l'intérieur de la même application ou vers d'autres applications (traitement de texte, ...).

Etant donné le nombre de références envisagées, l'utilisation de l'outil informatique pour atteindre ces objectifs paraissait s'imposer, par sa rapidité et l'éventail de ses possibilités.

L'absence sur le marché d'un tel produit impliquait un développement local : l'idée de **GABI APA Style** venait de naître ...

II. DÉVELOPPEMENT DU LOGICIEL

Le développement de GABI s'est organisé autour de quelques principes généraux qui concourent tous au même objectif : créer un logiciel non seulement utile, mais qui ait toutes les chances d'être utilisé.

Je souhaitais un **logiciel complet et performant**. Un logiciel qui permette de gérer plusieurs milliers de références et autorise leur présentation selon des **normes officiellement reconnues**, mais qui, tout en restant rapide, offre également un maximum d'autres potentialités d'exploitation, dont au moins celles que l'on retrouve habituellement sur un système moderne de gestion de bases de données. Pour cela, une attention particulière devait être accordée à la **structure des fichiers et index**.

Cette performance ne devait pas être antinomique avec la facilité d'utilisation. Il fallait que GABI soit un système **convivial**, agréable à utiliser et "intelligent", pour qu'il puisse être utilisé aisément par quiconque, sans connaissances particulières préalables, ni en informatique, ni en normes bibliographiques.

Bien que ces deux premiers termes supposent le développement d'un outil de taille conséquente, je voulais que malgré tout il puisse fonctionner sur la majorité des machines habituelles. Ceci qui impliquait en particulier, un travail important sur **l'utilisation de la mémoire vive**.

Enfin, ce système, ne pouvait être fermé sur lui-même, il devait pouvoir **communiquer** avec d'autres applications comme des traitements de texte, mais éventuellement aussi d'autres systèmes de gestion de bases de données.

Tous ces éléments font la matière des chapitres suivants. Mais avant tout, il est nécessaire de dire quelques mots des **choix initiaux** qui ont du être faits quant aux normes bibliographiques à utiliser et quant au langage de programmation. De la pertinence de ces choix dépendrait en effet dans une large mesure la possibilité d'atteindre les objectifs.

A. CHOIX INITIAUX

1. CHOIX DE LA NORME BIBLIOGRAPHIQUE DE L'A.P.A.

Aucune norme bibliographique n'est universelle et qui plus est, toutes sont entachées d'un certain nombre de limites. En choisir une a toujours quelque chose de restrictif et est sujet à controverse. On pourra toujours objecter que telle autre lui eut été préférable. Mais les arguments développés pour soutenir l'un ou l'autre choix ne sont souvent le fait que d'habitudes personnelles ou d'écoles.

Les éléments qui ont présidé au choix des normes de l'A.P.A. (American Psychological Association) sont eux, purement d'ordre pragmatique et ne veulent pas entrer dans ce débat.

- **Au niveau de la Faculté**, les rares personnes qui utilisaient de manière systématique des normes pour formater leurs références employaient celles de l'A.P.A.;
- **Au niveau international**, ce sont celles qui sont le plus largement utilisées dans le domaine de la psychologie : tous les périodiques de l'A.P.A. et un nombre important d'autres périodiques exigent des auteurs la présentation de leurs sources bibliographiques sous cette forme. Un système qui permettrait la production aisée de références selon ces normes serait donc d'un intérêt indéniable pour le chercheur.

En ce qui concerne les sciences de l'éducation, aucune norme ne s'y est clairement imposée tant qu'à présent. Celles qui existent connaissent une diffusion limitée. L'application des normes de l'A.P.A. n'y pose par contre aucun problème, sinon d'ordre autonomiste. D'aucuns dans la Faculté s'en inspirent d'ailleurs plus ou moins fidèlement.

Les descriptions **ISBD**, très largement utilisées dans le domaine des bibliothèques, avaient l'avantage de dépasser le cadre limité de la psychologie et d'être beaucoup plus complètes et précises que la plupart des normes de citations de références.

Cependant, elles ont pour principal objectif la description catalographique du document par le bibliothécaire. Elles sont par conséquent encombrées de détails inutiles pour l'utilisateur scientifique final. Elles sont également plus limitées dans leurs possibilités d'exploitation concrète par le scientifique, car leur usage effectif ne dépasse guère le cadre des bibliothèques.

De plus, à un niveau plus épistémologique, on peut dire que les normes ISBD abordent le document en tant qu'objet matériel, tandis que les normes de citations de références bibliographiques le considèrent dans son contenu, en tant que thématique développée par un auteur. Cette dernière position est beaucoup plus proche des besoins et des préoccupations du chercheur.

a) SOUCI D'EFFICACITÉ ET DE SOBRIÉTÉ

Les règles bibliographiques de l'A.P.A. ont été élaborées pour standardiser la présentation des références dans les publications de l'A.P.A. Ce qui les caractérise, c'est leur souci d'efficacité. Leur objectif est uniquement de permettre l'identification rapide et univoque du document. Elles éliminent donc les détails inutiles dans ce cadre, pour ne présenter que les éléments indispensables.

- Ainsi, la numérotation des fascicules de périodiques ne doit pas être indiquée, sauf lorsque la pagination de chaque fascicule à l'intérieur d'une année reprend à 1 (dans ce seul cas en effet, le volume et les pages sont insuffisants pour retrouver de manière certaine le document).
- Dans la même perspective, ces règles exigent de mettre en évidence par des caractères italiques ou soulignés, non pas l'information la plus importante en soi (ce devrait être alors le titre du document), mais celle qui permet d'orienter le plus rapidement la recherche du document (le titre pour les monographies, le titre du périodique pour les articles de revues, ...).
- De même, l'année de publication est placée très tôt dans la référence (juste après les auteurs), contrairement à d'autres normes. Cela permet de retrouver très rapidement une référence citée dans le texte puisque la citation s'y fait au moyen du nom de l'auteur suivi de la date de publication et que les listes de références doivent être classées dans l'ordre alphabétique des auteurs, avec un sous-classement par date de publication.
- Elles évitent autant que possible les abréviations pour éliminer les équivoques et rendre la référence immédiatement intelligible.

Enfin, d'aucuns regretteront que des éléments qu'ils considèrent importants ne soient pas repris : le format du document, la présence d'illustrations ou le nom d'un éventuel préfacier par exemple. Ces éléments principalement de type descriptif, sont cependant superflus par rapport à l'objectif poursuivi.

b) DESCRIPTION SUCCINCTE

Les documents subissent un traitement particulier en fonction de leur appartenance à une des grandes catégories suivantes : articles de périodiques, ouvrages, articles d'ouvrage avec éditeur scientifique, thèses de doctorat, actes de congrès, colloques ou symposium, rapports techniques ou de recherche, documents non publiés.

Pour chacune de ces catégories, la référence est découpée en un certain nombre de champs. Certains de ces champs sont communs à toutes les catégories (auteurs, titre), avec parfois des nuances (date d'édition pour la plupart des documents; de défense pour les thèses de doctorat); d'autres sont spécifiques à une seule catégorie (titre de périodique, numéro de rapport) ou à plusieurs (maison et ville d'édition).

Chacun de ces champs se termine par un point qui agit comme séparateur et permet d'identifier rapidement les différents éléments de la référence.

Pour chaque champ, l'A.P.A. indique les règles de présentation normale et précise par des exemples, l'attitude à adopter face aux cas particuliers. Bien que ces cas particuliers

soient inévitables, ils complexifient singulièrement la structure générale qui elle est relativement simple.

c) LIMITES

Les normes bibliographiques de l'A.P.A. sont décrites dans un ouvrage précisant l'ensemble des règles stylistiques exigées des auteurs proposant une publication à l'A.P.A. Y sont également repris les consignes de présentation générale du texte, les formes exigées pour les titres, résumés, tableaux, figures, notes de bas de page, formulations mathématiques. Les consignes concernant les marges, ponctuations, abréviations, capitalisations, ... y sont également indiquées.

Dans ces conditions, il est difficile d'être exhaustif en un peu plus de vingt pages. Un certain nombre de cas particuliers restent non résolus. Ainsi, rien n'est dit des mémoires de licence par exemple. Mais il est clair qu'aux yeux de l'A.P.A., ce type de document est d'un intérêt très limité. Une adaptation aux mémoires des règles concernant les thèses doctorales est cependant aisée.

Elaborées par la très importante association américaine des psychologues, on comprend qu'on y observe un certain américano-centrisme.

Enfin, comme toute règle, elles ont quelque chose de contraignant et de rigide dont il faut s'accomoder (pourquoi la pagination d'un article d'ouvrage avec éditeur scientifique doit-elle être précédée de "pp." alors que pour un article, les pages extrêmes sont données seules ?).

2. NÉCESSITÉ DE DÉVELOPPEMENT D'UNE APPLICATION SPÉCIFIQUE

Un tour d'horizon des logiciels disponibles a démontré l'absence ou l'inadéquation de produits permettant de répondre aux objectifs fixés. Il était par conséquent impossible de faire l'économie du développement local d'un logiciel spécifique.

a) INADÉQUATION DE LIBER © DANS CE CONTEXTE PARTICULIER

L'utilisation de l'application documentaire LIBER dans ce contexte particulier ne paraissait pas judicieuse pour plusieurs raisons.

- La principale a trait aux objectifs mêmes. LIBER a été installé dans le réseau des Unités de Documentation et Bibliothèques de l'Université de Liège pour gérer les documents **effectivement** possédés par celles-ci. Or, l'objectif de la base de données à créer était de rassembler l'information sur l'ensemble des publications des membres de la FAPSE, qu'elles **soient ou non disponibles** dans l'U.D., voire à l'Université Il est clair que la volonté de constituer parallèlement un fonds de ces documents devrait à terme réduire cette discordance, sans pour autant qu'il soit possible d'escompter la faire disparaître. Il n'en reste pas moins que les finalités demeurent différentes.

- La politique du réseau des bibliothèques et U.D. de l'Université concernant la saisie sous LIBER est d'introduire prioritairement les ouvrages entiers, les thèses, ainsi que les titres des collections possédées par l'Université. Bien que certains responsables d'U.D. le fassent d'ores et déjà, y introduire des articles de périodiques, des chapitres d'ouvrages, voire des rapports de recherche non publiés ... et cela en dehors d'une politique systématique, ne pourrait qu'encombrer la base de données actuelle et la rendre hétéroclite.
- Une autre raison quasi aussi importante, est que LIBER ne permet pas de formater les références selon les normes de l'APA, ce qui constituait également un objectif principal. Il n'est pas interdit de penser qu'il eut été possible de développer sous PICK des modules permettant ce type de présentation à l'affichage et à l'impression, mais d'une part l'investissement en programmation n'aurait sans doute pas été plus faible qu'en Clipper, d'autre part, ce sous produit à destination d'une seule Faculté aurait risqué d'encombrer le système pour les autres utilisateurs.
- Les possibilités d'exportation et d'importation sous LIBER sont actuellement inexistantes du point de vue de l'utilisateur. Or je souhaitais un outil ouvert, dont les produits puissent être immédiatement récupérés pour d'autres applications (traitement de texte, ...). Cette caractéristique me semblait même être une condition essentielle de son utilisation effective.
- Enfin, les critères de recherche disponibles avec LIBER semblaient trop limités par rapport à ce qui serait utile dans ce contexte spécifique.

b) ABSENCE DE PRODUIT PERMETTANT DE GÉRER DES BASES DE DONNÉES AVEC LES NORMES DE L'A.P.A.

Lorsque la décision de développer GABI a été prise, le seul produit qui existait sur le marché et permettait de gérer directement des références selon les normes de l'A.P.A. était le traitement de texte "*Manuscript manager APA Style*"[™] développé par Pergamon Press © et qui prend en compte toutes les règles éditoriales de l'A.P.A.²

Cependant, les limites de ce produit en ce qui concerne la gestion de références ne sont pas négligeables :

- La création d'une référence nécessite d'avoir auparavant acquis une bonne connaissance des normes de l'A.P.A.;
- Le cadre est celui d'un traitement de texte et non pas d'un outil de gestion de base de données. Bien qu'il soit possible de créer des bibliothèques de références dans lesquelles on peut puiser celles dont on a besoin dans un texte particulier, il n'existe aucune possibilité de recherche même primitive. Il faut balayer l'ensemble de la bibliothèque, ce qui devient vite fastidieux et peu efficace dès que le nombre de références augmente.

²

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.

- Le système a manifestement été conçu pour un utilisateur individuel ayant un nombre limité de références à gérer. Il est difficile d'imaginer son comportement si on l'utilisait pour introduire quelques centaines voire quelques milliers de références, même en faisant abstraction de l'impossibilité de faire des recherches.
- Il s'agit d'un système fermé. Il n'existe aucune possibilité d'échange de données en dehors du logiciel. Or le traitement de texte est relativement élémentaire par rapport à ce que l'on trouve actuellement sur le marché.

Lorsqu'on développe un logiciel, la terre ne s'arrête pas de tourner. Parallèlement à la mise au point de GABI, d'autres concepteurs ont fait le même constat de manque d'outils permettant de gérer des références selon des normes établies et se sont attachés à en élaborer.

Ainsi j'ai découvert tout récemment (mai 1992) l'existence d'un tel produit, appelé "*Reference Manager*" © développé par Research Information Systems™. Ce logiciel disponible sur Macintosh® et sur PC présente un nombre important de fonctionnalités communes avec GABI, ce qui est rassurant du point de vue de l'analyse du problème. Il présente également des potentialités supplémentaires intéressantes, mais également des désavantages non négligeables :

- Dans la première catégorie, on peut citer principalement la possibilité d'**élaborer soi-même des formats d'affichage** des références en paramétrant l'ordre et la présentation des éléments. Ce logiciel ne gère pas une norme particulière comme celle de l'A.P.A., mais se veut adaptable à toute norme. Le revers de la médaille est évidemment qu'il faut une connaissance préalable approfondie des normes pour permettre la paramétrisation du logiciel et que les cas particuliers sont négligés.
- Un autre point fort de cette application est qu'elle permet des **liens** aisés avec certains traitements de texte. La différence entre la technique des liaisons et celle des exportations et incorporations telle qu'elle est prévue dans GABI, est que dans le premier cas, les modifications apportées dans la base de données se répercutent immédiatement dans les textes auxquels les références sont liées.
- Enfin, elle est d'ores et déjà disponible en version **Macintosh** et **PC** compatible. Sans en avoir fait le test, je suppose que les échanges de données entre ces deux environnements est possible.

Du côté des inconvénients, on peut relever que :

- ce logiciel ne gère que les cas "faciles" (ouvrages entiers, articles de périodiques, chapitre d'ouvrages ...). Rien n'est prévu pour les thèses de doctorat, rapports de recherche, actes de congrès, ... ;
- même dans ces cas, la paramétrisation ne permet pas de traiter les cas particuliers dont on a vu qu'ils étaient particulièrement nombreux. Paramétrée selon les normes de l'A.P.A. par exemple, la présentation de la référence n'est que partiellement correcte;
- il n'y a que peu de contrôle automatique à la saisie (il est par exemple possible de donner une année, un volume et même des numéros de pages à

un article sous presse l) et l'assistance au moyen des fichiers d'autorité n'est pas automatique;

- les seules informations complémentaires qu'il soit possible d'ajouter sont des descripteurs. Rien ne permet d'imaginer que l'exploitation de relations hiérarchiques entre descripteurs soit envisagée dans les développements futurs du logiciel;
- les possibilités de recherche sont limitées à quelques possibilités (auteur, année, titre de périodique, descripteur). Il n'y a pas de création automatique d'un dictionnaire de tous les mots-clefs, ce qui est pourtant particulièrement utile;
- quelques essais avec un Macintosh puissant et sur une base d'une septantaine de références seulement semblent indiquer que la recherche n'est pas extrêmement rapide. Il n'est pas certain que ce logiciel puisse supporter des centaines, voire des milliers de références en conservant des délais d'attente acceptables;
- il n'existe pas de possibilité d'importation/exportation sur disque des références en format ISO 2709, ce qui constitue pourtant un standard international en matière de communication de références entre machines. Il n'est donc malheureusement pas possible d'échanger des références entre GABI et ce logiciel;
- l'ergonomie du logiciel est très largement améliorable;
- enfin, mais cela est secondaire, je ne suis pas certain qu'il soit disponible en français.

Sur tous ces points, GABI APA Style se démarque positivement. De plus, il est spécifiquement adapté au contexte particulier de la FAPSE. Il garde donc tout son intérêt. Cependant, il n'est pas impossible que d'autres logiciels plus concurrentiels encore apparaissent. Ce sera à l'utilisateur de faire le choix en définitive, et d'utiliser l'outil qui lui paraît le mieux adapté à son besoin spécifique.

3. CHOIX DE CLIPPER TM

a) CARACTÉRISTIQUES ET AVANTAGES

Pour développer ce gestionnaire automatisé de bibliographies, il m'a semblé opportun d'utiliser Clipper comme langage de programmation. Les raisons qui ont motivé ce choix, sont les possibilités de ce langage dans le domaine de la gestion des bases de

données, mais aussi l'expérience que j'en avais acquise par le développement d'une autre application conséquente ³.

Clipper TM est avant tout un compilateur dBASE TM extrêmement performant produit par Nantucket Corporation ®. C'est en cela qu'il s'est fait connaître à l'origine en 1985, car il permettait de faire fonctionner sous forme compilée et donc beaucoup plus rapidement, des procédures écrites en dBASE. Certains informaticiens ne l'utilisent d'ailleurs toujours que dans ce cadre restrictif.

Cependant, Clipper est beaucoup plus qu'un simple compilateur. Bien plus même qu'un simple langage de programmation orienté bases de données. On peut le considérer comme un véritable environnement de programmation. Il se démarque d'ailleurs de plus en plus du dBASE d'Ashton-Tate ® et tend à devenir indépendant.

En effet, 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 : en particulier en ce qui concerne la gestion de l'environnement, les variables, les champs, la manipulation de tableaux, les index, le nombre de fichiers ouverts simultanément.

De plus, outre la palette des commandes et fonctions dBASE, il ajoute en standard beaucoup d'autres possibilités extrêmement puissantes.

Il permet aussi d'exploiter de très nombreuses et très complètes bibliothèques de fonctions spécialisées disponibles sur le marché ⁴, en particulier *Clipper Tools One* TM de Nantucket que j'ai largement utilisé. Il est même possible de créer ses propres commandes et fonctions en langage C et de les incorporer en Clipper. Les commandes et fonctions de bases de Clipper sont d'ailleurs écrites en C.

Avec ces outils complémentaires, il est aisément possible de sortir du cadre limité de la gestion de bases de données pour développer à peu près n'importe quel type d'application.

Enfin, par rapport à d'autres, c'est un langage de haut niveau. Il se caractérise par l'existence de commandes et fonctions extrêmement puissantes. Dans bien des cas, il suffit de quelques lignes de codes pour réaliser des opérations particulièrement complexes

b) PRINCIPALES LIMITES

A côté de ces avantages, Clipper présente un certain nombre de limites.

Les principales concernent la taille très importante des fichiers EXE créés au moment du "link". Ceci s'explique par le fait que la philosophie de la compilation de Clipper est en

³ Développement en Clipper d'un outil informatisé d'évaluation des capacités cognitives de planification : TICAPLAN.

⁴ Il existe ainsi des bibliothèques spécialisées de fonctions statistiques ou économiques, d'applications graphiques, de gestion des périphériques, elles comprennent également de très nombreuses possibilités de manipulations des chaînes de caractères et nombres, de gestion du système, de larges possibilités de communications via l'interface série ...

fait d'intégrer l'ensemble des ressources en commandes et en fonctions dans le module exécutable, contrairement à d'autres principes de compilateur qui ne traduisent que les commandes et fonctions réellement utilisées dans le programme source. Ainsi, si on compile avec Clipper un pseudo programme d'une seule ligne de code : une commande de nettoyage d'écran par exemple, on obtient néanmoins un module exécutable long de pas moins de 157 Ko !

Or, malgré l'évolution actuelle du hardware, même la dernière version de Clipper ne permet pas d'utiliser la mémoire étendue. La seule forme de mémoire au-delà de la mémoire conventionnelle que la version 5 de Clipper permet d'exploiter est la mémoire expansée de type LIM (Lotus-Intel-Microsoft). Or ce type de mémoire est moins répandu que la mémoire haute et étendue. De plus, même dans ce cas, la gestion est automatique et ne peut être modulée par le développeur.

On est donc forcé, lorsque l'on développe des applications un tant soit peu conséquente en Clipper version *summer 87*, de s'atteler à la création d'overlays pour réduire la taille du programme en mémoire. Sans quoi, on risque d'atteindre la fatidique limite des 640 Ko et d'empêcher le programme de fonctionner, surtout si l'on utilise quelques programmes résidents.

Il faut espérer que l'ouverture amorcée dans la version 5 vers les compléments de mémoire s'accroîtra dans les prochaines versions de Clipper et qu'elles permettront d'accéder à la zone de mémoire haute pour ouvrir les fichiers par exemple, et à la mémoire étendue pour faire fonctionner l'application (y placer les variables par exemple).

Un autre inconvénient de Clipper est en fait une conséquence du précédent. L'opération de compilation, mais surtout celle de "link" est très longue. Elle laisse presque le temps de boire un café. Ceci peut s'avérer pénible en phase de débogage, car la moindre faute de frappe à un endroit quelconque du programme, implique, après correction, une nouvelle compilation pour pouvoir poursuivre le test ... Or l'excès de café est nuisible pour la santé !

Enfin, Clipper est prévu initialement pour fonctionner en mode texte, ce qui est limitatif au niveau de l'affichage. Si l'on souhaite utiliser le mode graphique, il faut utiliser des bibliothèques de fonction supplémentaires. Ceci agrandit encore la taille du module exécutable et limite alors l'accès à certaines fonctions conçues pour le mode texte.

c) CHOIX DE LA VERSION

GABI a été développé avec la version *Summer 87* de Clipper.

Le lecteur pourra s'étonner de cette utilisation d'une version de Clipper qui ne soit pas la dernière parue. Trois raisons sous-tendent ce choix :

- Tout d'abord, le développement de GABI a duré près de 2 ans et lorsque début 1992, le *Clipper 5* a fait son apparition sur le marché et est devenu accessible, l'avancement du produit était tel qu'une mise à jour permettant de profiter des nouvelles fonctionnalités de la version 5 aurait occasionné un retard très important.
- De plus, bien que cette version ait été annoncée comme présentant une compatibilité totale avec l'ancienne, ceci n'est en fait valable que pour les commandes et fonctions de la version de base. La bibliothèque

d'outils *Clipper Tools One*, produite pourtant également par Nantucket, est quant à elle incompatible avec le *Clipper 5*. Il faut la remplacer par sa mise à jour onéreuse et qui ne m'était pas accessible.

- Enfin, les seuls apports de la nouvelle version Clipper qui à priori me semblaient réellement intéressants par rapport au développement de GABI et auraient pu me conduire à une adaptation du programme, concernaient la gestion de la mémoire : en particulier la possibilité de créer des overlays dynamiques. Or, après expérimentation, il a fallu constater que si la création de ces overlays est singulièrement facilitée, ils ont plutôt tendance à ralentir l'exécution du programme par rapport à des overlays statiques bien organisés.

Il est probable que les développements futurs de GABI, demanderont cependant l'exploitation des potentialités nouvelles qu'offriront les mises à jour successives de Clipper, et surtout des bibliothèques de fonctions qui ne manqueront pas de fleurir dans cet environnement.

B. ELÉMENTS DE LA RÉFÉRENCE

1. AMÉNAGEMENTS PAR RAPPORT AUX NORMES DE L'A.P.A.

Dans sa version actuelle, GABI gère toutes les règles générales du style A.P.A. pour les principales catégories de référence. Un certain nombre d'aménagements ont cependant été apportés par rapport à la norme A.P.A. stricte, principalement en ce qui concerne les cas particuliers et les exceptions.

a) GESTION DES CAS PARTICULIERS ET EXCEPTIONS

Dans la description des normes bibliographiques de l'A.P.A. , la place qui leur est réservée, est beaucoup plus importante que celle consacrée aux règles générales. Certaines règles correspondent d'ailleurs à des cas tout à fait exceptionnels dans un contexte habituel.

Vouloir les inclure de manière systématique dans GABI aurait encombré inutilement le logiciel. Aussi, ai-je choisi comme politique de négliger les règles qui ne présentaient pas une probabilité suffisante d'apparition dans le cadre des documents publiés par des membres de la FAPSE. Le cas échéant, l'utilisateur pourra la plupart du temps adapter sa référence à un des cas prévus.

Principaux cas non traités actuellement par GABI :

- titre dans une langue autre que l'anglais, dont on souhaite donner une traduction en anglais dans la référence;
- 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;
- la plupart des documents non publiés à l'exception des thèses de doctorat et des mémoires de licence;
- support non imprimé tel que film, enregistrement audio, logiciel informatique,...;
- article de magazine ou quotidien;
- "review "d'article.

Bien entendu, ce choix reste empirique et pourra être revu s'il s'avère qu'un cas non traité actuellement par GABI se présente de manière significative. Ainsi, il apparaît dès à présent que les deux derniers cas devront être incorporés à la prochaine version du logiciel.

b) AUTRES AMÉNAGEMENTS

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

De même, en ce qui concerne les thèses de doctorat non publiées, la mention exacte "*unpublished doctoral dissertation*" a été remplacée par son équivalent français auquel j'ai ajouté le domaine (de manière surprenante, les normes de l'A.P.A. ne l'exigent pas). 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, je les ai incorporés car ce type de document reste crédité d'une certaine valeur par les membres de la FAPSE et par les utilisateurs de l'U.D. Je leur ai appliqué une adaptation des règles propres aux thèses (mention : "*mémoire de licence en sciences de l'éducation non publié*"). J'ose espérer que les puristes n'en seront pas offusqués.

2. INFORMATIONS COMPLÉMENTAIRES

En plus des éléments de la référence A.P.A., un certain nombre d'éléments pour la plupart facultatifs ont été ajoutés. Ces informations complémentaires permettent d'aller au-delà de la simple citation bibliographique et, par l'analyse qu'ils permettent, d'enrichir les possibilités d'exploitation de la base de données en en faisant un véritable outil de documentation.

- le **service d'origine** : le nom du service éventuel au sein duquel le document a été élaboré;
- la **langue** dans laquelle le document est écrit. 11 langues sont proposées. Si l'utilisateur n'y trouve pas celle qui l'intéresse, il lui reste la possibilité de se rabattre sur le choix "autre langue". Il n'est pas impossible qu'avec l'Europe et l'ouverture des pays de l'Est, des possibilités supplémentaires devront être offertes dans une version ultérieure (!);
- la **cote de rangement** du document (dans la perspective de la constitution à l'U.D. d'un fonds de tous les documents publiés par des membres de la FAPSE);
- les **descripteurs**. 20 descripteurs peuvent être liés au document. Dans la version actuelle, ces descripteurs doivent être choisis au moyen d'un thesaurus externe. Il n'est donc pas possible actuellement d'exploiter leurs relations génériques pour permettre par exemple l'explosion d'un terme sur ses termes spécifiques ou la recherche guidée par les termes associés. Ces fonctionnalités sont prévues pour la version 3 de GABI.

A ces éléments optionnels, GABI ajoute :

- l'**identificateur de la base** : suite de 6 caractères attribuée par le gestionnaire de la base lors de sa création et qui correspond à l'identité de la base en cours;
- le **numéro d'identification de référence** : il s'agit d'une numérotation monotone attribuée par le système lors de la création de la référence et qui constitue sa clef d'accès. Joint à l'identificateur de la base il devient l'étiquette de la référence qui la suivra dans toute son histoire;
- 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 peuvent ou non figurer à l'affichage, l'impression ou le downloading.

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

C. STRUCTURE DES FICHIERS ET INDEX

Un des deux principes directeurs du développement de GABI a été de construire un outil qui soit **performant**, c'est-à-dire qui permette de très larges possibilités d'exploitation tout en restant suffisamment rapide ⁵. Ceci implique une structuration complexe des fichiers dont on trouvera un aperçu ci-dessous.

Avec les fichiers d'autorité, les fichiers relationnels et tous les index et sans compter les fichiers temporaires de travail, c'est près de 70 fichiers qui sont ouverts simultanément lorsque GABI fonctionne.

1. BASES DE DONNÉES RELATIONNELLES

GABI APA Style a été développé selon le principe des bases de données relationnelles.

a) PRINCIPE

On considère généralement qu'il existe deux types principaux de bases de données, selon leur mode d'organisation des fichiers :

(1) LES BASES DE DONNÉES CLASSIQUES

Ce sont les plus connues du grand public. C'est généralement des fichiers de ce type que la plupart des utilisateurs créent avec les outils de bases de données habituels (dBASE, FOX TM ...). Elles placent l'ensemble de l'information dans un seul fichier. Celui-ci comprend autant de champs qu'il peut y avoir d'éléments différents dans les enregistrements.

⁵ On pourrait épiloguer sur le problème de la "vitesse" d'une application informatique. Aujourd'hui, la plupart des utilisateurs trouve insupportable toute attente de quelques secondes devant l'écran, alors que les mêmes opérations faites manuellement prendraient parfois de nombreuses heures de travail soutenu et auraient nécessité il y a quelques années seulement de longues minutes de travail à un ordinateur de l'époque.

Le degré de rapidité "suffisant" doit donc être compris ici à un niveau subjectif : c'est le délai que la plupart des utilisateurs sont prêts à accepter pour l'opération. Dans la pratique, excepté pour les grosses opérations ce délai ne dépasse généralement pas 1 seconde.

Num enre	Auteur1	Auteur2	...	Titre	Année	Périodiques	Edit. Comm	...
12	Richelle	de Keyser	...	Etude de la ..	1987		Mardaga	...
23	De Keyser			Sécurité et ergonomie	1989	Cahiers de
75	Crahay	Richele		Epistémologie...	1990		Masson	...
...

L'avantage de ce type de système, est qu'il est assez simple à concevoir et à manipuler. Il peut même être utilisé dans le mode interactif initialement prévu par l'outil de gestion de base de données (dBASE, ...), ce qui permet d'éviter la programmation. Cependant, il souffre de plusieurs inconvénients importants :

- **il consomme énormément de place disque inutile** car tous les champs possibles doivent être définis pour tous les enregistrements, même s'il ne sont pas utilisés.

Par exemple, s'il est envisageable de rencontrer des références comportant 15 auteurs, il faut prévoir 15 champs auteurs, même si la plupart des références n'en utiliseront que 1 ou 2 ! Ou alors, on se contente de stratégies de recherches séquentielles, très lentes par définition. De même, il faut créer des champs pour tous les éléments spécifiques de tous les types de référence, (par exemple un champ "*Titre de Périodique*" qui sera vide pour toutes les références de monographies, des champs pour les "*éditeurs scientifiques*" pour les articles d'ouvrages avec éditeurs scientifiques, ...).

Le nombre total de champs et partant, la longueur totale d'un enregistrement, deviennent vite très élevés, bien que la part effectivement utilisée pour une référence particulière soit nettement plus limitée. Pour un fichier reprenant les éléments de GABI, on peut estimer que 10 à 20 % de l'espace disque nécessaire seraient réellement utilisés, soit une perte globale supérieure à 80 % !

- **il ne permet pas aisément de vérification à la saisie.** A chaque enregistrement, les données sont introduites sans aucun contrôle par le système qui ne sait pas en vérifier le caractère éventuellement redondant.

Par exemple, on peut introduire le nom d'un auteur sous des formes différentes, même avec des fautes de frappe, sans que le système ne réagisse (dans l'exemple illustré : de Keyser/De Keyser ou Richelle/Richele). Ce type d'erreurs de saisie inévitables aura pour conséquence immédiate de diminuer l'efficacité de la recherche.

(2) LES BASES DE DONNÉES RELATIONNELLES

Le principe en est tout différent. En simplifiant, on peut dire que la référence n'est pas sauvegardée d'un seul tenant. Elle est découpée en de multiples éléments, enregistrés de manière séparée.

Pour chaque type d'information dont les éléments présentent un caractère répétitif d'une référence à l'autre (auteurs,...) et pour lequel on souhaite une possibilité de vérification à la saisie, on crée une base séparée, appelée **fichier d'autorité**. La base principale ne comporte dès lors plus que les éléments non répétitifs. Dans chaque fichier d'autorité, on n'introduira les éléments qu'une seule fois, ce qui permet d'éviter tous les problèmes de graphies multiples. Entre la base principale (appelée **fichier maître**) et chaque fichier d'autorité, on crée une base dite "**relationnelle**" qui permet d'établir la relation : de spécifier quels éléments du fichier d'autorité sont liés à un enregistrement du fichier maître.

FICHER MAITRE

Num enre	Titre	Année	...
12	Etude de la ..	1987	...
23	Sécurité et ergonomie	1989	...
75	Epistémologie...	1990	...
...

FICHER RELATION.

Num enre	Num aut
12	1
12	2
23	2
75	3
75	1
...	...

FICHER AUTORITE

Num aut	Auteurs
1	Richelle, M.
2	de Keyser, V.
3	Crahay, M.
...	...

Cette structure qui éclate la référence en de nombreux fichiers, outre l'aide importante qu'elle permet à la saisie, rentabilise beaucoup plus efficacement l'espace disque en évitant les redondances (gain de place estimé : de 60 à 80 %). Elle est cependant beaucoup plus lourde à concevoir et à gérer. En effet, chaque fois qu'il faut présenter l'enregistrement, il est nécessaire de le reconstituer à partir des différents fichiers d'autorité. De même, les mises à jour et modifications devront être correctement répercutées sur l'ensemble des fichiers concernés.

Etant donné les avantages du principe des fichiers relationnels, la plupart des applications professionnelles de gestion de bases de données fonctionnent selon celui-ci.

b) APPLICATION À GABI

A côté du fichier maître, 9 fichiers d'autorité ont été construits, munis chacun de leur fichier relationnel correspondant :

Les trois premiers sont communs à l'ensemble des références :

- auteurs;
- descripteurs;
- mots clefs.

Les 4 suivants sont spécifiques à un ou plusieurs types de documents :

- villes et pays de publication;
- éditeurs commerciaux;
- titres de périodiques;
- universités où la thèse ou le mémoire a été défendu.

Enfin, les 2 derniers ont un statut particulier :

- liste des **pays connus** (ce fichier ne sert qu'à faciliter la saisie. Pour de simples raisons d'efficacité, il n'y a pas de retour à ce fichier lors de la reconstitution de la référence);
- **anti-dictionnaire multi-lingue** (ce fichier sera abordé en page 27).

Ces fichiers d'autorité impliquent la création de différents index permettant de voyager de n'importe quelle référence du fichier maître vers les différents éléments correspondants et inversement pour les procédures de recherche.

2. CHAMPS

a) CHAMPS À LONGUEUR FIXE ET À LONGUEUR VARIABLE

Clipper est conçu pour fonctionner avec des **champs de longueur fixe**. C'est-à-dire qu'il faut déterminer au départ la dimension du champ en fonction de la taille de l'enregistrement le plus long qu'il risque de devoir contenir et préférentiellement, en s'accordant encore quelques espaces de réserve. Dans tous les cas où l'élément à y introduire est plus court, la place disponible est remplie par des espaces et donc perdue.

Ainsi, même si la plupart des noms d'auteurs ne dépassent pas une quinzaine de lettres, initiales des prénoms comprises, il faut au minimum prévoir une longueur de 40 caractères pour ce champ. Sinon un nom tel que *Van Den Keerbergerheid-Bergerelles*, *A-M* risquerait d'être irrémédiablement tronqué par exemple, de même qu'une collectivité-auteur, ...

leur **ordre d'apparition** et en majuscules pour éviter les problèmes de classement alphabétique sur les caractères accentués.

A l'exception du premier auteur, les initiales des prénoms ont été éliminées car le risque d'erreur de classement à ce niveau est négligeable. De même, la longueur totale du champ a été calculée pour permettre d'y inclure une dizaine d'auteurs car le risque de rencontrer deux références ayant plus de 10 auteurs exactement identiques est également très faible.

Par contre, il est envisageable de rencontrer deux références ayant exactement les mêmes auteurs et publiées la même année. Dans ce cas, le classement se fait sur le titre. Or les champs à longueur variable de type mémo ne permettent pas d'indexation. Il a donc fallu créer un champ à longueur fixe reprenant les premiers caractères du titre du document pour permettre cette indexation. Ce nombre de caractères pouvait être assez limité. En effet, même si dans certains pays la règle d'or est "*publish or perish*", il est peu fréquent de rencontrer deux références des mêmes auteurs publiés la même année et dont les titres commencent par un nombre important de mots identiques.

Cependant, un tel cas n'est pas exclu, par exemple lorsqu'un même document est publié en 2 parties, distinctes uniquement par le sous-titre. Ce cas n'a pas été rencontré dans l'échantillon de 609 références de membres de la FAPSE introduites jusqu'à présent, mais reste imaginable. GABI y réagirait en les plaçant dans l'ordre où elles ont été introduites. Ceci devra peut-être faire l'objet d'un réaménagement ultérieur, toujours en tenant compte d'un équilibre entre l'efficacité et le degré de redondance et partant, de consommation d'espace disque ...

Signalons enfin que l'image du début du titre n'est pas une simple reproduction des premiers mots du titre. Elle a subi une double transformation : mise en majuscules et typographie pauvre pour éviter les problèmes de classement sur les caractères accentués et suppression de l'article initial sur lequel ne porte pas le classement. Ceci limite encore la redondance.

3. INDEX PRÉÉTABLIS

Pour permettre le classement des enregistrements, Clipper est prévu pour fonctionner avec un système d'index ⁶, c'est-à-dire des fichiers de pointeurs distincts. Cependant, rien n'interdit d'adopter la politique des fichiers inversés, tels qu'ils fonctionnent dans le système STAIRS © d'IBM ® par exemple et de développer les structures correspondantes, en exploitant les possibilités offertes par les champs mémo de Clipper.

La technique des champs inversés a l'avantage de rendre les fonctions de recherche extrêmement rapides. Par contre, elle implique deux inconvénients massifs : d'une part, elle impose de très lourdes opérations de mise à jour de ces fichiers, opérations qui ne sont pas raisonnablement possibles en temps réel, ce qui a toute une série de conséquences : maintien en parallèle d'une base de données effective et de fichiers de mise à jour, impossibilités d'exploiter les références introduites depuis la dernière opération de mise à jour, ...). D'autre part, et c'est également une conséquence de la première remarque, elle implique une consommation disque très importante.

⁶ Il est intéressant de noter que les indexes créés par Clipper sont du type "B-tree branching", beaucoup plus rapides que les indexes créés en dBASE.

On comprendra que pour ces raisons, l'établissement d'index qui permettent quant à eux des mises à jours immédiates, a été préféré au développement de fichiers inversés.

En ce qui concerne ces index, 2 stratégies générales étaient envisageables :

- soit ne définir au départ que les index strictement nécessaires au fonctionnement du système (ceux liés aux fichiers d'autorité et aux fichiers relationnels) et ne construire qu'à posteriori, chaque fois que nécessaire, les autres index;
- soit construire au départ l'ensemble des index qui pourront être utilisés et les mettre à jour en temps réel.

C'est cette seconde politique qui a été préférée pour GABI. Elle consomme assurément beaucoup plus d'espace disque et nécessite l'ouverture simultanée d'un plus grand nombre de fichiers, ce qui coûte également de la mémoire vive. Par contre, elle permet d'éviter tout délais lors du passage d'un type de classement à l'autre, et est par conséquent beaucoup plus confortable ⁷. Le ralentissement à la saisie est lui quasi insensible.

Cependant, toujours dans le but d'optimiser la vitesse et l'efficacité, ce principe souffre une exception. Lorsque le nombre de références à communiquer suite à une recherche est restreint, le balayage systématique et complet du fichier maître dans l'ordre de l'index choisi occasionnerait des délais importants. Ceux-ci seraient inversement proportionnels au rapport entre le nombre de références à communiquer et le nombre de références total du fichier. Dans ce cas, il a été jugé préférable de provoquer la construction d'un fichier temporaire, d'y reproduire les références nécessaires à partir du fichier maître puis de créer un index temporaire sur cette base restreinte. Cette technique améliore sensiblement les temps de réaction dès que le nombre de références à communiquer est inférieur à 10 % de l'ensemble du fichier.

4. FICHIER DES MOTS-CLEFS ET ANTI-DICTIONNAIRE MULTI-LINGUE

Une des fonctionnalités de GABI est de permettre la recherche de références à partir de mots-clefs, c'est-à-dire de n'importe quel mot significatif de la référence ou du champ descripteur. Lors de la saisie, ces mots sont ajoutés automatiquement par le logiciel, à un fichier d'autorité appelé **dictionnaire**.

Pour éliminer les mots non-significatifs qui alourdiraient inutilement ce dictionnaire, 3 techniques sont utilisées dans GABI :

- élimination des mots de moins de 2 lettres (rares sont les mots de moins de deux lettres sémantiquement forts);
- élimination, le cas échéant, de la pagination et du numéro de volume;
- confrontation avec un anti-dictionnaire multi-lingue.

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

Il n'a pas paru réaliste ni intéressant de tenter de développer un outil syntaxique permettant d'éliminer les variations flexionnelles des mots, d'autant que dans la plupart des cas, l'utilisation judicieuse de la troncature y remédie aisément.

L' **ANTI-DICTIONNAIRE** est constitué sous forme d'une base de données des mots non significatifs les plus fréquents et de leurs dérivations syntagmatiques. Cependant, certains mots sans signification dans une langue peuvent avoir un homographe porteur de sens dans une autre langue (le démonstratif allemand *diese* et le substantif français *dièse*; la préposition néerlandaise *met* et le substantif français *met* du langage culinaire; la préposition allemande *wegen* qui signifie à cause de et le verbe néerlandais *wegen* : peser, le pronom relatif allemand *deren* qui signifie dont et le verbe néerlandais *deren* qui signifie nuire ...). Par conséquent, il m'a semblé utile de rendre cet anti-dictionnaire multi-lingue : ne seront éliminés que les mots vides de la langue d'appartenance du document.

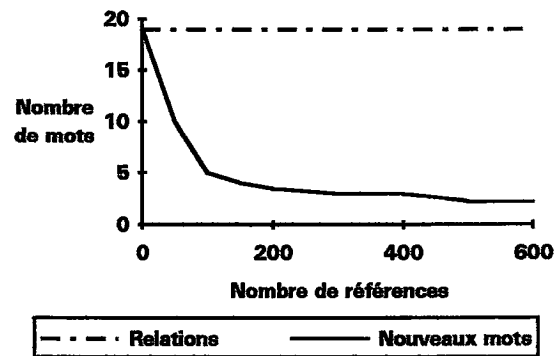
Ceci ne résout pas le problème plus fréquent encore des homographies à l'intérieur d'une même langue (en français : la conjonction *mais* et le substantif *mais*; la conjonction *car* et le substantif *car* : autocar; la préposition *vers* et le substantif lui-même polysémique *vers*; la préposition *sous* et le substantif pluriel un peu vieilli *sous* signifiant argent, ...). Dans de tels cas, la structure de décision d'inclusion ou non du terme dans l'anti-dictionnaire est basée sur un principe pragmatique. Si l'acception ayant une signification a des chances d'être rencontrée plus souvent que de manière exceptionnelle dans le contexte de références scientifiques du domaine de la psychologie et des sciences de l'éducation, le terme n'est pas inclus dans l'anti-dictionnaire. Il vaut mieux en effet encombrer quelque peu le dictionnaire de quelques termes sans signification, plutôt que de perdre la possibilité de faire des recherches sur un terme qui pourrait s'avérer intéressant.

Cet dictionnaire négatif multi-lingue comprend au départ près de 1100 mots appartenants aux 7 langues les plus couramment utilisées (français, anglais, néerlandais, allemand, italien, espagnol et portugais). Il n'est pas fermé : il est possible de le compléter avec d'autres termes d'une des 11 langues connues de GABI, à travers le module de gestion des fichiers d'autorité.

Enfin, pour terminer, une remarque à propos de la taille du fichier dictionnaire et du fichier relationnel associé. Avec leurs index correspondants, il s'agit des plus gros fichiers mis en place par GABI. Ceci ne pose pas trop de problèmes pour le fichier dictionnaire lui-même car sa croissance suit une courbe logarithmique (la première référence ajoutée dans la base de données implique la création d'environ 19 mots dans le fichier dictionnaire, mais, lorsque 600 références ont déjà été incorporées, toute nouvelle référence n'ajoute plus en moyenne que 2,2 mots nouveaux au dictionnaire). Ceci est par ailleurs une démonstration manifeste de l'intérêt des structures relationnelles de fichiers par rapport aux bases de données classiques.

Par contre, la taille du fichier relationnel et de ses index augmentent de manière linéaire (plus de 34000 enregistrements pour une base de 1800 références, ce qui signifie qu'en moyenne 19 mots significatifs sont traités par GABI pour chaque notice).

Outre l'espace disque que cela nécessite ⁸, l'importance de ces fichiers est telle qu'elle peut rendre impossible une opération lourde telle que la réorganisation des fichiers, lorsqu'on utilise un système disposant d'une mémoire vive limitée.



⁸ Avec une base de donnée de 1800 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 subit récemment une réorganisation.

D. CONVIVIALITÉ ET ERGONOMIE LOGICIELLE

Le deuxième souci majeur qui a présidé au développement de GABI est la volonté de créer un outil agréable et facile à utiliser, même et surtout par un utilisateur peu formé en informatique. Ceci impliquait un travail au niveau de la couche **superficielle** (gestion des écrans, menus déroulants, assistance à la saisie, aide en ligne "*context specific*" ...) mais aussi et surtout à un **niveau plus profond** et partant, moins visible (souplesse du système, procédures de vérification, gestion des erreurs, logique de l'utilisateur ...).

Le travail sur la convivialité et l'ergonomie d'un logiciel a quelque chose de profondément frustrant. Le but est en effet de rendre transparent et d'une apparente facilité, voire même, évidence, des structures sous-jacentes parfois singulièrement complexes. Le clinicien qui sommeille en moi ne pouvait pas y voir moins qu'une ... blessure narcissique !

1. NIVEAU SUPERFICIEL

a) GESTION DES ÉCRANS

Une attention toute particulière a été consacrée à la construction des écrans de manière à permettre une lecture aisée et un repérage rapide de l'information nécessaire à un moment donné.

Pour y arriver, plusieurs techniques ont été utilisées :

- **multi-fenêtrage** : l'affichage est organisé autour d'un écran de base sur lequel s'ouvrent en superposition, avec une illusion de relief et éventuellement en cascade, différentes fenêtres, en fonction de l'opération effectuée. Ce système fréquemment utilisé dans les logiciels actuels permet de renforcer l'impression de continuité et d'aider l'utilisateur à suivre le fil de sa propre démarche à travers la "trace" des écrans précédents qui subsistent en arrière-plan.
- **menus déroulants** : cette technique est devenue habituelle dans les logiciels modernes. Dans GABI, elle fonctionne de pair avec le multi-fenêtrage. Sur l'écran de base ne sont repris en une seule ligne que les titres des grandes familles de fonctions disponibles. Il suffit de positionner la surbrillance sur un de ceux-ci pour qu'apparaisse, "*se déroule*" le menu complet des possibilités offertes dans ce domaine. Dès que l'on souhaite passer à une autre famille, le premier menu disparaît et c'est le suivant qui se déroule.
- **divisions de l'écran en zones fonctionnelles** : l'affichage d'informations de même type se fait toujours dans la même zone de l'écran, afin de faciliter le repérage : l'utilisateur sait vers où il doit regarder pour trouver une information qu'il attend.
 - La ligne supérieure constitue la première zone : la "*barre-menu principale*", réservée aux grandes familles de fonctions de GABI;

- dans la partie supérieure de l'écran s'affichent soit les **sous-menus principaux** soit les **messages** (aide ou erreur);
- dans la moitié inférieure apparaîtront les **résultats** (résultats d'une recherche ou résultat d'une opération réalisée par le système, tel qu'une étape d'installation, ...);
- la dernière ligne de l'écran ou d'une fenêtre est réservée au **rappel des touches actives**.
- **standardisation des attributs d'écran (choix des couleurs,...)** : quel que soit la procédure en cours, on retrouve les mêmes couleurs pour des opérations ou des informations sémantiquement similaires, ces informations contextuelles facilitent et accélèrent l'interprétation du message (noir sur rouge signifie message d'erreur, zone verte : zone où il faut répondre quelque chose à GABI, caractères rouges : information importante, clignotement : opération en cours, ...).

b) ASSISTANCE À LA SAISIE

(1) SIMPLIFICATION DE LA SAISIE

L'utilisateur n'a pas à se préoccuper lors de la saisie de la formalisation de la référence selon les règles de l'A.P.A.

GABI ne présente que les éléments ayant un sens pour le type de document en cours d'introduction. De même, il gère automatiquement les mentions spécifiques (ajout de "Ed." ou "Eds." de "pp."... chaque fois que nécessaire), l'ordre des éléments, la ponctuation, les parenthèses, ...

Bien que certains logiciels similaires tel que *Reference Manager* ne réagissent pas ainsi, tout ceci est suffisamment évident pour permettre d'éviter de s'y attarder.

(2) UTILISATION DES LISTES D'AUTORITÉ

Que ce soit en mode recherche, ajout ou modification d'une référence ou gestion des fichiers d'autorité, chaque fois que l'utilisateur doit introduire un terme contrôlé par un fichier d'autorité (auteur, titre de périodique, ville d'édition, pays d'édition, université où a été défendue une thèse ou un mémoire, descripteur), le logiciel opère une triple vérification en cascade :

1. il vérifie d'abord si le terme n'existe pas dans le fichier d'autorité exactement sous la forme introduite;
2. sinon, il vérifie s'il possède des enregistrements comportant comme racine l'élément introduit par l'utilisateur;
3. sinon, il vérifie s'il ne possède pas d'enregistrements ayant au moins les premières lettres communes avec cet élément;

4. sinon enfin, il signale à l'utilisateur qu'il ne connaît pas cet élément et, soit lui propose de l'ajouter, dans le cas de la saisie ou modification, soit de le réintroduire à nouveau.

Si l'utilisateur préfère se contenter d'introduire quelques lettres seulement (cas N° 2) ou fait une erreur dans la frappe au clavier (cas N° 3), GABI réagit en lui proposant une liste des possibilités les plus proches dans laquelle il n'y aura plus qu'à choisir l'item adéquat.

Cette aide à la saisie, permet de l'accélérer sensiblement et d'éviter facilement les erreurs.

(3) AUTO-CORRECTIONS

La capitalisation des mots dans les normes de l'A.P.A. suit des règles complexes et d'une logique toute relative (les initiales de tous les mots significatifs du titre d'un périodique doivent être mises en majuscules, par contre, pour le titre du document, à l'exception des noms propres, seule la première lettre du premier mot du titre et du sous-titre - bien que celui-ci soit précédé de deux points - doit être mise en capitale, les noms des auteurs ne peuvent être mis en majuscules à l'exception de leurs initiales ...).

Afin d'éviter à l'utilisateur la mémorisation de ces règles avec le risque d'erreurs que cela comporte, GABI accepte toutes les formes à la saisie pour les transformer automatiquement, en les rendant conformes à la norme A.P.A.

De même, par exemple dans le cas de l'introduction du nom de l'université dans laquelle un mémoire ou une thèse a été défendu, si le GABI constate que l'utilisateur a omis d'indiquer le terme "université" ou une traduction, il l'ajoute lui-même.

(4) VÉRIFICATION DE LA COHÉRENCE INTERNE

Lorsque faire se peut, GABI vérifie la plausibilité des données introduites par l'utilisateur.

Par exemple il examine la date de publication et la refuse si elle est postérieure à l'année en cours. Il refuse aussi une page de fin d'article plus petite que la page de début d'article, ... Il empêche l'introduction d'une thèse de doctorat ou d'un mémoire de licence anonyme (!) ou muni de plusieurs auteurs, ...

c) AIDE EN LIGNE "CONTEXT SPECIFIC"

Outre les messages d'information qui constituent une aide élémentaire et qui apparaissent à l'écran lors des différentes opérations, il existe un système d'aide détaillée sur l'utilisation du logiciel. Ce système est 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 différentes. Chacune de celles-ci se présente en 1, 2, 3 ou 4 écrans successifs.
- soit en pressant la touche F1 à tout moment où GABI attend une action de l'utilisateur (saisie d'information, choix dans une liste, ...). Apparaît alors dans une fenêtre superposée, la rubrique d'aide correspondant à l'opération en cours. Si on le souhaite, il est alors possible de voyager dans les autres rubriques d'aide puis de revenir à l'étape restée en cours.

Ces écrans d'aide, d'une part donnent une explication sur l'opération en cours et d'autre part précisent la liste des touches fonctionnelles à ce moment et leur effet respectif. Ils devraient permettre de se passer quasi de tout support écrit pour utiliser le logiciel avec succès.

d) RECHERCHES PRÉ ET POST-COORDONNÉES

GABI offre les deux possibilités classiques d'accès aux données, ainsi que toutes les combinaisons possibles entre elles. Selon les besoins de l'utilisateur, ou sa connaissance préalable d'éléments des documents qu'il recherche, il lui est possible de faire des recherches :

- **pré-coordonnées**, par titre de périodique, maison d'édition, université ou descripteur ... L'utilisateur connaît au départ l'élément qu'il recherche et l'introduit directement sous sa forme exacte (par exemple, il sait qu'il s'agit d'un titre de périodique et en connaît avec certitude le nom)
- **post-coordonnées**. Dans ce cas, l'utilisateur sait uniquement qu'il cherche des références qui touchent à un thème, mais n'a aucune idée de l'endroit de la référence où ce thème pourrait apparaître. Il fera sa recherche au moyen de mots-clefs qu'il lui suffira de combiner au moyen des opérateurs booléens.

Ceci permet même, par exemple de rechercher les documents publiés dans un périodique dont on n'avait au départ qu'une idée vague du nom.

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.

e) PARAMÉTRISATIONS

GABI a été conçu de manière à ne pas être un produit rigide et fermé. L'utilisateur peut paramétrer à sa guise un certain nombre d'options des modules de communication en

fonction de ses préférences personnelles ou de la tâche qu'il a à réaliser (affichage, impression, exportation normale ou condensée, ou avec des sous-titres, plusieurs possibilités de classement, plusieurs formats d'exportation, possibilité de personnalisation du titre principal, ...). Dans de nombreuses autres procédures également, des choix ont été prévus pour s'adapter aux souhaits de l'utilisateur (mode d'évaluation de la nécessité de réorganisation, installation, statistiques, ...).

D'autres paramétrisations auraient pu être ajoutées, comme par exemple un choix personnalisé des couleurs, ainsi que cela est disponible sur beaucoup de logiciels actuels. Cependant, cela me paraissait être plutôt de l'ordre du gadget que de l'apport d'un plus significatif.

Par contre, il serait intéressant, dans la perspective d'un élargissement du produit, d'étudier la possibilité d'offrir le choix entre d'autres normes de présentation des références (avec éventuellement le format I.S.B.D. réduit). Il faudrait évaluer le bénéfice qu'en retireraient les utilisateurs par rapport à l'alourdissement des opérations de saisie et de stockage qui en résulterait.

Enfin, signalons que 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 (recherche, affichage, impression, exportation, statistiques, ...). Les opérations de gestion et de maintenance sont protégées par un système de mot de passe encrypté et réservées au gestionnaire de la base. Celui-ci peut à volonté modifier son mot de passe pour des raisons de confidentialité.

2. NIVEAU PROFOND

a) LOGIQUE DE L'UTILISATEUR

Lors du développement de GABI, un soin particulier a été accordé à l'adaptation du logiciel aux processus cognitifs spontanés de l'utilisateur et accentuer de manière systématique sa transparence. 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 aux structures cognitives des utilisateurs.

Ce travail a porté essentiellement sur :

- le **regroupement des fonctions en familles** présentant un lien conceptuel et l'organisation de l'ordre des familles sur la barre-menu principale en disposant les éléments prioritaires ou susceptibles d'être le plus fréquemment utilisés, en fonction du balayage gauche-droite;
- la **modularisation du système** : découpage en unités simples correspondant en unités logiques qui se succèdent en cascades. Un exemple en est la gestion de la recherche booléenne. Ceci permet à l'utilisateur de ne traiter qu'un problème particulier à la fois, 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 lorsque l'utilisateur désire créer un fichier sur disque (installation ou exportation), il n'a pas à se préoccuper de savoir si le chemin spécifié existe ou s'il doit être créé. Il lui suffit d'indiquer celui qu'il souhaite et, si nécessaire, GABI réalise de manière transparente toutes les opérations de découpage et de création des répertoires emboîtés;
- **le traitement des erreurs** : lorsque l'utilisateur sollicite le déroulement d'une opération erronée, le système l'en avertit au moyen d'un message explicatif, puis le ramène à l'étape initiale (par exemple : recherches ou statistiques sur une base ne contenant encore aucune référence, recherche d'une étape non encore réalisée, effacement d'étapes de recherches inexistantes, modification d'une référence inexistante, installation sur une unité de disque virtuel, dans un chemin impossible, ...);

Dans le cas d'une sollicitation inattendue, GABI demande une confirmation préalable (communication de l'ensemble de la base en cours, effacement d'un grand nombre de références, ...);

- **la gestion des touches actives** : d'une part, prise en compte de l'effet attendu de l'utilisation des touches de déplacement par rapport à la représentation spatiale que le sujet se fait du système, d'autre part, standardisation de l'utilisation de certaines touches. L'usage de la touche < F1 > pour l'appel d'aide est devenu classique, la combinaison < CTRL > < END > permet en mode saisie, d'effacer une proposition de GABI.

Le cas de la touche < ESC > est intéressant. Conformément à beaucoup de logiciels actuels, une "signification" d'échappement à la procédure en cours et de retour à l'étape antérieure en abandonnant les 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 ultimes, en ayant à tout moment la possibilité de revenir au branchement directement supérieur. Pourtant, cette représentation est illusion : elle n'est que partiellement conforme à la structuration interne de l'application. De plus, il faut souligner combien, au niveau profond, les effets sont en fait différents même s'ils ont une apparence d'identité de signification.

Enfin, dans un certain nombre de cas, plusieurs possibilités parallèles permettent indifféremment d'atteindre le même résultat. Pour se déplacer dans le barre-menu principale par exemple, on peut utiliser les flèches ou les initiales majuscules des grandes fonction. Ceci est une première approche de la mise en place de **touches de raccourcis** pour les utilisateurs avertis.

Précisons toutefois qu'aucune procédure expérimentale n'a été mise en place pour vérifier le bien-fondé des choix effectués. Cependant, trois techniques ont été utilisées à cet effet :

- **décentration** : à toute étape du développement les décisions ont été prises en ayant systématiquement à l'esprit des questions du type *"Comment à ce moment, l'utilisateur réagirait-il logiquement ? A quoi s'attendrait-il ? Quelles réactions du système escompte-t-il en pressant telle touche ? ..."*.

- **mise en relation** avec les observations réalisées dans un autre cadre : l'étude des stratégies cognitives de planification : par exemple : technique des sujets de découpage d'un problème complexe en unités simples successives; apparition parasite de contraintes ajoutées en fonction de la représentation de l'espace problème que se fait le sujet, ...
- **observation à posteriori**, lors de la phase de β -test de sujets placés en situation, analyse de leurs réactions et remarques. En fonction du matériel récolté, rectifications du logiciel.

b) PROCÉDURES DE VÉRIFICATION DES DOUBLONS

Ces procédures automatiques ont été introduites afin d'éviter l'introduction involontaire de doublons dans la base de données.

Lors de la saisie d'une nouvelle référence, dès qu'un nombre suffisant d'éléments ont été introduits, GABI vérifie si une référence vraisemblablement identique n'a pas déjà été introduite auparavant. Si tel est le cas, il en avertit l'utilisateur en temps réel, lui présente la référence "soeur" et l'interroge pour savoir s'il souhaite malgré tout poursuivre l'introduction (il peut s'agir d'une pseudo-égalité). L'apparition de la moindre divergence dans la comparaison des références conduit GABI à les considérer comme différentes. Ceci peut occasionnellement être insuffisamment discriminant en cas de faute de frappe dans le titre par exemple, ou saisie d'une référence déjà introduite, mais avec des auteurs introduits dans un ordre différent.

Une vérification similaire s'opère lors de l'importation de références, si du moins l'utilisateur a sélectionné l'option d'importation avec vérification. Cependant, pour ne pas forcer l'utilisateur à stationner devant l'écran pendant toute la durée de l'importation, celle-ci se fait en mode automatique et GABI crée un fichier ISO des éventuels doublons présents dans le fichier importé. Le traitement de ces doublons se fait en temps différé. A la fin de la procédure, il indique à l'utilisateur le nombre de doublons vraisemblables qu'il a rencontrés. Libre alors à celui-ci de les importer ultérieurement, un par un et en mode manuel, à partir du fichier d'identiques ainsi créé.

c) ENREGISTREMENT DIFFÉRÉ

Lors de la saisie d'une référence, il peut arriver pour différentes raisons que l'utilisateur souhaite corriger ou abandonner les éléments qu'il a déjà introduits. Pour cette raison, l'enregistrement effectif des données sur le disque est différé. Il se fait en phase finale, lorsque tous les éléments ont été introduits et que l'utilisateur a pu vérifier la forme que prendra en normes A.P.A la référence qu'il a introduite. A ce moment seulement, il peut décider l'enregistrement effectif global ou la correction de l'un ou l'autre élément.

De plus, ceci permet à tout moment, d'interrompre la saisie sans risque d'altération des fichiers.

Cette technique a comme avantage supplémentaire d'accélérer la saisie en supprimant les temps de latence intermédiaires nécessaires aux enregistrements successifs sur disque.

d) SOUPLESSE

GABI s'adapte sans intervention de l'utilisateur, à l'environnement hardware dans lequel il fonctionne. Ses attributs d'écran se sélectionnent automatiquement en fonction de la carte contrôleur d'écran qui est installée (monochrome ou couleur) et les appréciations de durée d'opération qu'il donne, sont fonction du processeur XT ou AT.

De même, s'il constate qu'il se trouve dans un état d'installation partielle (version de transport), il limite les possibilités d'exploitation aux seules options opérationnelles à ce moment. En particulier, il engage l'utilisateur à installer une version complète avant de rendre accessibles les autres opérations.

e) 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 dans lequel il est implanté. Si un problème survient, il en avertit l'utilisateur avec un message explicite lui expliquant la source du problème et si possible, le moyen de le surmonter.

Dès le démarrage, il vérifie la version du système d'exploitation installé et la présence des fichiers indispensables à son fonctionnement. Il vérifie également si le système est configuré de manière à permettre d'ouvrir un nombre de fichiers suffisant. Le cas échéant, il affiche un message d'erreur, et, selon la gravité du problème, permet la poursuite de l'application ou revient au DOS.

De manière permanente, GABI vérifie aussi la quantité de mémoire vive disponible. Si celle-ci devient insuffisante pour certaines opérations, il en empêche le déroulement, avertit l'utilisateur et lui conseille de libérer de la mémoire, par exemple en éliminant des programmes résidents.

De même, il surveille le bon déroulement de chaque opération d'entrée/sortie (I/O) que ce soit sur disque ou sur imprimante. Il analyse les problèmes éventuels (imprimante non branchée ou non allumée, manque de papier, nom d'unité invalide, disquette non introduite, disquette protégée en écriture, manque de place sur la disquette ou le disque, chemin d'accès invalide, ...) et envoie à l'utilisateur un message lui expliquant la source de l'erreur et le guidant dans les mesures de correction à effectuer.

E. GESTION DE LA MÉMOIRE

Dès que le développement du logiciel a pris une certaine ampleur ⁹ et que le nombre de modules et de procédures est devenu important, je me suis trouvé confronté au problème de l'espace mémoire exigé pour faire fonctionner GABI.

Dans sa version actuelle en utilisant pourtant le mode de compilation le plus économique mais sans segmentation de la mémoire, GABI occuperait plus de 635 Ko de mémoire vive. Le programme à lui tout seul friserait donc la limite fatidique des 640 Ko de mémoire conventionnelle exploitables sous DOS. Et cela, sans compter les 130 à 250 Ko nécessaires pour stocker les variables, tableaux, résultats de recherche,... Sans compter non plus la place occupée par le système (fichiers systèmes, gestionnaires de périphériques, programmes résidents, table d'environnement, ...) !

Or, il a été rappelé en page 15, que Clipper version *summer 87* ne permet pas l'exploitation de mémoire ni haute, ni étendue, ni paginée. Seule la mémoire conventionnelle peut être utilisée pour les applications. D'ailleurs, même si la tendance actuelle des constructeurs est d'installer en standard de tels compléments de mémoire, tous les micro-ordinateurs habituellement rencontrés n'en disposent pas encore. De plus, un des objectifs du développement de l'application était de permettre son fonctionnement quel que soit, ou presque, la configuration hardware du PC.

Comme tel, le programme ne pouvait donc en aucun cas fonctionner. Un travail important a du être entrepris dans plusieurs directions pour réduire à un strict minimum, la dimension en mémoire de l'application .

1. RATIONALISATION DES LIGNES DU CODE

La première technique utilisée, la plus évidente, fut un travail de rationalisation du code source de manière à le dépouiller de tout élément superflu :

- **standardisation** des opérations similaires et **élimination des redondances** par rassemblement des opérations récurrentes en procédures communes;
- réécriture de blocs de codes en développant des **algorithmes plus concis** et si possible plus performants. Dans ce cadre, diverses stratégies et structures de programmation ont été évaluées empiriquement pour déterminer les plus performantes et les plus synthétiques;
- suppression et **intégration** dans les procédures mères de petites procédures utilisées une seule fois et créées à seule fin de clarification de structure (avec une quantité de code identique, un découpage en procédures plus nombreuses "coûte" plus d'espace mémoire car chaque procédure occupe une position en mémoire et doit être repérée par une "clef" d'accès).

⁹ L'ensemble des programmes sources constituant GABI APA Style version 2.00 dépasse les 24 000 lignes de codes, soit près de 500 pages A4 de programmation.

2. STRUCTURATION EN OVERLAYS

a) PRINCIPLE

La deuxième technique consiste à l'utilisation d'overlays. Le principe des overlays est de ne placer en mémoire vive, outre les éléments de base indispensables, que les ensembles de procédures dont le programme a effectivement besoin à un moment donné. Les autres modules sont stockés sur le disque dur. Chaque fois qu'il en a besoin, le programme opère un "swapping", un échange : les premiers éléments sont éliminés de la mémoire pour faire place aux seconds. Une représentation éclairante de ceci est l'image d'une glissière : les overlays parallèles "coulissent" de gauche à droite de manière à amener l'élément adéquat dans la zone mémoire.

Cette technique permet des économies de mémoire appréciables, mais a deux inconvénients :

- elle implique un **découpage et une structuration** du programme extrêmement soignés en modules indépendants. Il ne faut pas en effet que le module en mémoire fasse appel à une procédure englobée dans un module non actif à ce moment. Sans quoi, il y a conflit d'accès à la mémoire. Pour cela, il faut séparer les procédures générales dont tous les modules - ou en tout cas plusieurs - peuvent avoir besoin, et les procédures spécifiques à un overlay particulier. Il faut aussi clairement distinguer les grandes fonctions de l'application indépendantes l'une de l'autre. De même, afin de maximiser les économies, il faut veiller à constituer des blocs de taille comparable. En effet, la taille mémoire que le programme doit réserver pour ces overlays est égale à celle du plus important d'entre eux, sinon, ce dernier ne pourrait se loger dans la niche réservée. Ce point nécessite une phase d'ajustements délicate.
- elle implique des **accès disques** chaque fois que le programme passe d'un overlay à l'autre. Le temps de chargement de l'overlay dépend de la vitesse d'accès de l'unité. Sur un disque dur rapide, le délai est quasi insensible, mais l'utilisation de l'application avec un système à disquette peut rapidement devenir problématique. Un autre facteur de ralentissement est lié à la structuration de l'application. Il faut étudier minutieusement les moments où l'overlay est appelé en mémoire pour éviter les accès disques inutiles.

b) OVERLAYS STATIQUES, OVERLAYS DYNAMIQUES

Ainsi qu'il a été expliqué en page 17, la technique d'overlay retenue pour le développement de GABI, est celle des overlays de type statiques.

La version 5 de Clipper offre pourtant l'énorme avantage de permettre la création automatique d'overlays dynamiques sans que le développeur ait à se préoccuper d'une quelconque manière du découpage de son application en unités logiques indépendantes. Ceci est pris en charge par Clipper lors de l'opération de "link".

Une série d'essais m'a pourtant amené à constater un ralentissement de l'application suite à l'utilisation de cette technique. Peut-être ceci s'explique-t-il par le fait que dans ce cas, les overlays créés par le système sont de taille extrêmement réduite : 2 Ko environ. Le système "switch" donc perpétuellement avec le disque dur. Une autre explication peut venir du fait que lorsque ces essais ont été réalisés, une part importante du découpage et de la structuration de l'application avait déjà été réalisée. Il n'est pas impossible qu'avec un programme développé en dehors de ce souci serait plus performant avec des overlays dynamiques.

c) APPLICATION À GABI : 3 SÉRIES ET 2 NIVEAUX

Une première application des overlays à GABI concerne les programmes liés aux écrans d'aide. Ceux-ci consomment près de 9000 lignes de programmation, ce qui aurait absorbé une large quantité de mémoire vive si la technique des overlays ne leur avait été appliquée. Afin de ne pas multiplier exagérément le nombre d'overlays, les quelques 100 programmes d'aide initiaux ont été regroupés en 25 unités constituant autant d'overlays. Lors de l'appel d'un écran d'aide via la touche < F1 > ou la fonction AIDE, le programme charge en mémoire uniquement le programme correspondant à la rubrique sollicitée. Ceci permet de ramener la place réservée aux fonctions d'aide à 11 Ko au lieu des 168 Ko qu'elles auraient nécessité autrement, soit un gain tout-à-fait appréciable.

Des overlays ont également été constitués pour les grandes fonctions du logiciel (les modules 1,2,3,4,...Y du schéma présenté plus bas). Au départ, par fonction, il fallait entendre les grandes familles de fonctions accessibles sur la barre-menu principale (recherche, gestion, communication, outils,...). Cependant, l'existence d'importantes disparités en taille de code entre ces différents groupes ¹⁰ m'a amené à un découpage plus détaillé dans certains groupes de fonctions, permettant de constituer des unités de taille comparable. De plus, pour permettre 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 afin de leur permettre des "glissements" parallèles et indépendants.

Pour équilibrer encore les tailles des overlays et réduire le code du module principal, toutes les procédures communes qui s'y trouvaient ont été analysées pour vérifier si elle ne pouvaient pas être basculées dans les modules les plus courts, libérant d'autant le module principal. De même, bien que structurellement participant à un autre niveau, un overlay reprenant la phase d'accueil et d'ouverture de l'application a été ajouté à cette série. Toujours pour réduire la taille du module principal, toutes les commandes d'initialisation des variables et de définition de l'environnement, opérations qui ne s'effectuent qu'une seule fois par session, y ont été incluses. En définitive, cette série comporte 9 blocs d'overlays.

En dernière analyse, il restait un module nettement plus important que les autres : le module gestion. Celui-ci ne pouvait être éclaté en différents modules isolés de plus petite taille. En effet, il existe une large interdépendance entre les procédures de ce module. Afin d'en réduire la place en mémoire, une troisième série d'overlays a alors été conçue. A la différence des 2 premières, celle-ci ne se situe plus au niveau principal, mais à un deuxième niveau : elle est emboîtée dans l'overlay GESTION. En simplifiant, on peut dire que ceci permet de ramener le code du module GESTION de près de 4000 lignes à un peu moins de 2000 lignes plus une zone mobile de 270 lignes. Ces overlays

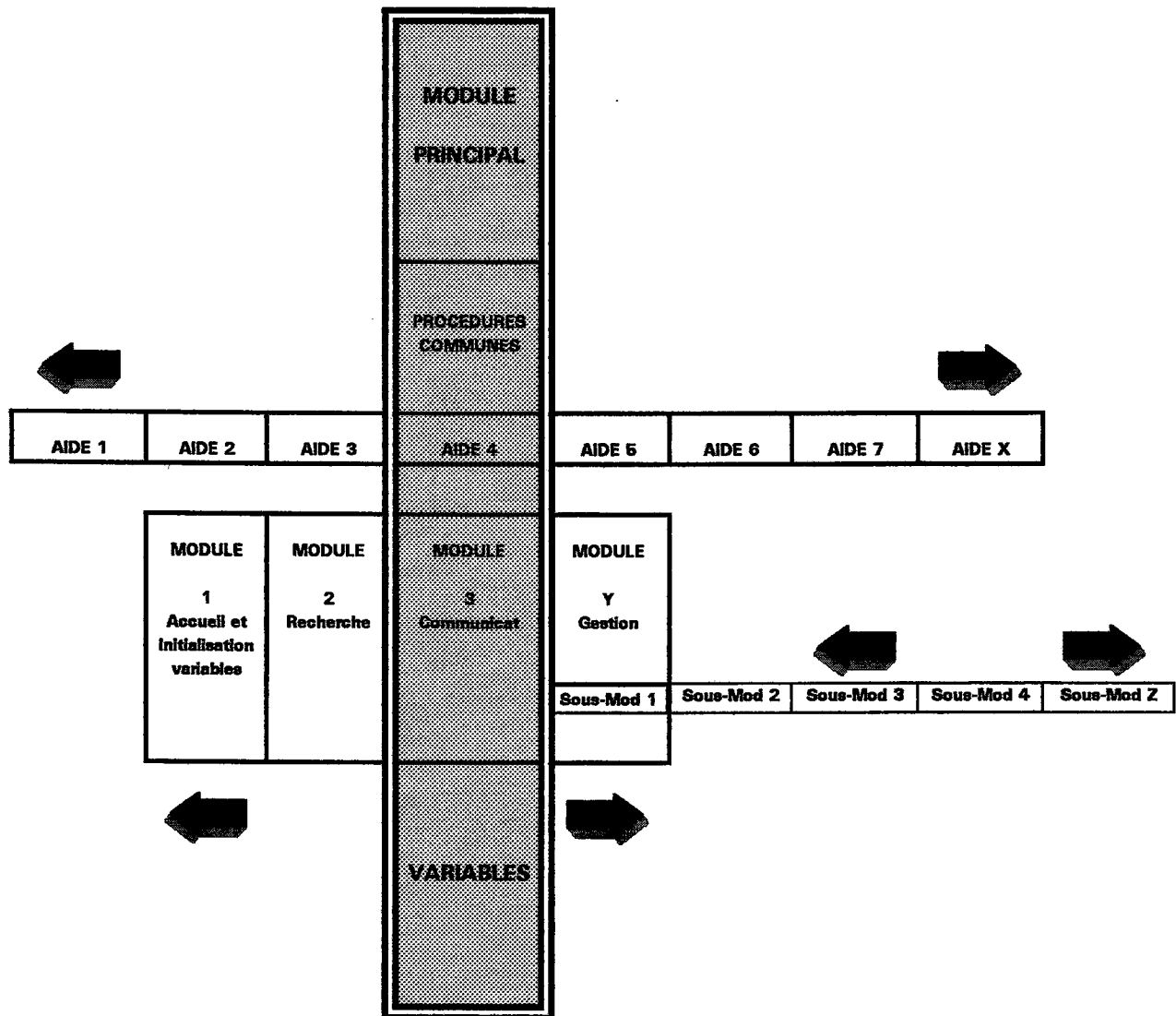
¹⁰ Le module OUTILS complet par exemple, prend plus de 4300 lignes de code, alors que le module COMMUNICATION n'en occupe que 1700.

imbriqués au module GESTION sont au nombre de 18. Ils apparaissent sous l'appellation "Sous-Mod" dans le schéma ci-dessous.

La mise en place de ces quelques 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, soit un gain appréciable de quasi 50 % (49,45 %).

Le lecteur trouvera ci-après une représentation graphique de la structure des overlays de GABI. Pour la facilité de lecture, tous les overlays d'un même niveau n'ont pas été représentés. Le grisé représente la place réellement occupée en mémoire vive avec les "niches" dans lesquelles "coulissent" les overlays des différentes séries.

**Représentation de la structure des overlays dans GABI APA Style :
3 séries en 2 niveaux**



On pourrait proposer une autre représentation imagée du programme et de ses procédures structurées en overlays : celle d'une usine et ses machines. Une technique pour réduire le volume nécessaire au bâtiment, est de ne garder provisoirement à l'intérieur que les machines dont on se sert à un moment donné. Les autres sont stockées à l'extérieur et ne sont rentrées que lorsqu'on en a effectivement besoin.

3. SAUVEGARDES TEMPORAIRES

Pour poursuivre la comparaison avec l'usine, lors de la conception des locaux, il faut aussi prévoir de la place pour ranger les matériaux : matières premières et produits finis. Les "boîtes" de stockage dans lesquelles on place ces matériaux, ce sont les variables du programme. Plus elles sont nombreuses et plus leur contenu est important, plus elles occuperont de place en mémoire. Dans GABI, pour permettre un fonctionnement strictement minimum, elles occupent 125 Ko supplémentaires.

2 types de variables occupent une place considérable en mémoire : celles permettant de conserver des écrans entiers (ce qui permet d'accélérer radicalement les nettoyages d'écran en évitant de les redessiner chaque fois qu'on y fait appel, mais coûte pas moins de 4 Ko par écran sauvé) et celles contenant les résultats de recherches ayant produit un nombre élevé d'occurrences. Malgré un codage résolument économique, le résultat d'une recherche ayant aboutit à plusieurs centaines d'occurrences, voire plus d'un millier, "coûte" plusieurs Ko.

Pour éviter que ces variables n'encombrent la mémoire suite à la multiplication de macro-recherches par exemple, je leur ai appliqué le principe de la sauvegarde temporaire. Il s'agit de stocker temporairement leur contenu sur disque et de ne le rappeler que lorsque nécessaire. Cette opération est systématique pour les résultats de recherche volumineux et occasionnelle pour les écrans (lorsque l'application va faire appel à une procédure consommant beaucoup de mémoire, comme par exemple la réorganisation complète des fichiers).

Ceci permet de rendre à nouveau disponibles à l'application, de larges quantités de mémoire. Dans ces conditions et en utilisant d'autres procédés plus techniques encore, l'application peut fonctionner à plein rendement et sans rencontrer de problème dans la plupart des conditions, en disposant de 200 à 230 Ko de mémoire disponibles en plus des 321 Ko nécessaires au chargement du programme.

En définitive grâce à l'utilisation de ces techniques de gestion de la mémoire vive, la taille de mémoire disponible nécessaire pour utiliser GABI est donc de 445 Ko au strict minimum et idéalement de 540 Ko. Ceci, en occasionnant des délais d'attente quasi imperceptibles, pour autant qu'un périphérique de stockage de masse rapide soit utilisé (disque dur et non disquette). Ces valeurs rencontrent l'objectif de permettre de faire fonctionner GABI sur une large majorité de configurations hardware habituellement rencontrées.

Néanmoins, il faut espérer que les versions ultérieures de Clipper permettront de gérer aisément la mémoire disponible au-delà des 640 Ko, ne fût-ce que pour y stocker les variables.

F. OUVERTURE DU LOGICIEL

Trop souvent, on rencontre des applications fermées sur elles-mêmes, et qui imposent de rester confiné dans leurs seules potentialités. Il en résulte par voie de conséquence, une mise en évidence de leurs limites. Le cas de *A.P.A. Manuscript Manager* est de ce point de vue éloquent. Aucun système, si complexe soit-il, ne peut se targuer d'être complet et performant dans tous les domaines ¹¹.

Une stratégie beaucoup plus efficace est assurément de développer des applications bien conçues mais ayant des objectifs clairement délimités, et d'offrir de larges possibilités d'interfaçage avec d'autres outils spécialisés dans d'autres domaines.

C'est dans ce souci d'ouverture que GABI a été conçu. Il permet deux types de communication. 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.

1. EXPORTATION EN FORMAT TEXTE

Il est possible de sélectionner en mode RECHERCHE un certain nombre de références puis de les exporter en conservant leur forme A.P.A. vers un fichier DOS ASCII, récupérable par un traitement de texte par exemple. Plusieurs formats d'exportation sont proposés :

- **format ASCII simple** : les références sont stockées selon les règles A.P.A. en caractères ASCII, donc sans aucune mise en forme des caractères. Le document peut être récupéré sous cette forme par la plupart des traitements de texte ou éditeurs de texte. L'absence de formatage ne permet pas de distinguer la partie qui doit 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 puis de créer une macro-commande permettant de rechercher ces repères et ainsi de repérer les zones à mettre en lettres italiques et de leur appliquer le format de caractère adéquat. Un exemple d'une telle macro est donné pour le traitement de texte WORD 5 de Microsoft ®;
- **formats particuliers** : un utilitaire de GABI offre la possibilité de créer des interfaces permettant la traduction des données et leur récupération directe par des traitements de texte spécifiques en ajoutant les codes de contrôles nécessaires pour le formatage. Ceci n'est cependant pas à la portée de tout

¹¹ Une telle ambition s'observe en particulier dans les logiciels intégrés. Elle aboutit presque toujours à un produit qui s'il n'est pas mauvais dans chacune de ces composantes, n'est cependant pas comparable, loin s'en faut, à de bons produits de chaque catégories. Ces intégrés peuvent néanmoins être intéressants pour des utilisateurs ayant des besoins limités et qui ne souhaitent pas investir dans l'apprentissage de plusieurs outils particuliers.

utilisateur, car cela nécessite une connaissance approfondie des codes de contrôle et des formats spécifiques des traitements de texte.

2. FOMAT D'ÉCHANGE ISO 2709

GABI permet également d'exporter et d'importer des références non plus en tant que texte fini, mais en conservant leur statut de données stockées dans des champs différenciés ¹². Bien qu'il eut été beaucoup plus simple de développer un format d'échange spécifique à GABI, j'ai préféré m'atteler à l'application du format ISO 2709, car il s'agit du canevas d'échange défini par l'*International Organization for Standardization* pour servir de standard dans les communications de données bibliographiques entre machines.

La possibilité d'exporter et importer des données dans ce format présente plusieurs avantages substantiels :

- permettre aisément 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 avec les index 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. Les logiciels importants comme STAIRS[®] ou LIBER[™] par exemple 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 différents composants de la référence, et le fichier créé par GABI devient lisible et exploitable par l'autre logiciel.

A titre exemplatif, voici comment se présente l'enregistrement d'une référence en format ISO créé par GABI.

```
00524N 0100196 34 001010000000201400100050020024007002002601000400280
200350032050017006705102600840520200110100017013015004601471700060193200019
01993000360218400028025445003702829000090319▲FAPSE 118▲010791/000000▲3▲F
▲▼AN▲▼Service de Psychologie du Langage▲▼HANDICAP MENTAL▲▼LANGAGE -- DEVE
LOPPLEMENT▲▼LANGAGE -- TROUBLE▲▼Rondal, J. A.▼1▲▼Language development and
mental retardation▼▲▼1987▲▼Oxford▼Angleterre▲▼Blackwell, Philadelphia Lippincott▲
▼W. Yule & M. Rutter (Eds.)▲▼Language development and disorders▼▲▼248-261▲↔
```

12

L'exportation en format texte expliqué plus haut est un produit achevé. Il n'est plus possible de découper automatiquement la référence pour y repérer les éléments constitutifs : différents auteurs, maison d'édition, date d'édition, titre du périodique, ...

Ce qui traduit en clair 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

G. DESCRIPTION DES PRINCIPALES FONCTIONNALITÉS DU LOGICIEL

Il ne sera donné ici qu'un bref aperçu des fonctionnalités principales du logiciel afin de permettre une vue d'ensemble. Pour une description détaillée, le lecteur se référera au **MODE D'EMPLOI** en annexe, ou le cas échéant, au texte des chapitres précédents.

Les fonctions sont décrites dans l'ordre de la barre-menu principal.

1. RECHERCHE

Des recherches sélectives peuvent s'opérer au moyen de **15 critères**. Certains de ceux-ci 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 dans la base en cours, ...), d'autres sont spécifiques à certains types de documents (titre de périodique, maison d'édition, ...).

Pour certains de ces critères, la recherche s'effectue avec l'aide des fichiers d'autorité correspondant (auteur, mot-clef, descripteur, titre de périodique, ville d'édition, université, ...). Pour d'autres qui ne disposent pas de tels fichiers, elle s'effectue par balayage séquentiel du fichier maître indexé (langue, année de publication ou défense, service d'origine, ...).

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

Les **combinaisons booléennes** sont possibles. Elles doivent être mono-nomiques à l'intérieur d'une séquence de recherche, pour éviter toute confusion. Pour réaliser des recherches complexes mêlant différents opérateurs, il suffit de découper l'équation en ensembles plus élémentaires ne comportant chacun qu'un seul type d'opérateur, puis de combiner de manière successive les résultats.

Par exemple, pour trouver le résultat de :

(A or B or C) and (D or (E not F))

On fera successivement :

étape 1 :	A or B or C
étape 2 :	E not F
étape 3 :	D or étape 2
étape 4 :	étape 1 and étape 3

Cette démarche par étape est plus longue, mais assurément plus aisée à interpréter pour l'utilisateur et moins sujette à des erreurs de saisie.

Lorsqu'une recherche a été réalisée, GABI indique le nombre d'occurrences qui correspondent à l'équation. Pour visualiser, imprimer ou exporter les références correspondantes, il suffit de passer dans le module communication.

2. GESTION

Cet ensemble de fonctions permet le développement de la base de données. Contrairement au module précédent qui était accessible à tout utilisateur, celui-ci est destiné au gestionnaire de la base de données. Chacune de ces fonctions est par conséquent protégée par mot de passe.

Il est possible d'ajouter des références, d'en modifier, d'en supprimer ou même d'importer via le format ISO des séries entières créées sur des systèmes GABI APA Style implantés sur d'autres machines.

L'ajout de nouvelles références est séquentiel. L'utilisateur est guidé et assisté par le logiciel de manière à simplifier la saisie et limiter au maximum 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 page 36). Dans le cas de l'importation, ce contrôle peut être déconnecté.

Enfin, lors de ces deux procédures, le dictionnaire des mots-clefs de la référence, accessibles à la recherche est élaboré automatiquement, avec confrontation avec un anti-dictionnaire multi-lingue et élimination des mots non significatifs (voir page 28).

3. COMMUNICATION

Les références sélectionnées au moyen des procédures de recherche, voire même l'ensemble des références que compte la base de données, peuvent être communiquées selon 3 canaux principaux : l'affichage à l'écran, l'impression ou le stockage sur disque.

Plusieurs paramètres de communication sont modulables : l'ordre des références, leur mode condensé ou non, la présence de sous-titres, le modèle d'imprimante, le format d'exportation, ... Ces éléments peuvent être modifiés dans le module des PREFERENCES de la rubrique OUTILS.

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

Outre la possibilité d'exportation en format texte récupérable par une application de gestion de texte, GABI permet l'exportation de références en format ISO permettant l'échange de données entre machines (voir page 44).

4. OUTILS

Cette rubrique compte huit d'utilitaires :

- le **module préférences** permet de paramétrer selon des préférences individuelles les 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. Enfin, il permet de changer le mot de passe du gestionnaire de la base de données;
- le **module fichiers d'autorité** permet des manipulations directes de ces fichiers. Etant donné les conséquences qu'une modification malheureuse pourrait avoir (les modifications portent sur l'ensemble des références utilisant l'élément concerné et peuvent être irréversibles dans les cas de transferts), ce module est également réservé au gestionnaire de la base et protégé par le mot de passe. En fonction du fichier d'autorité, diverses opérations peuvent être réalisées : corrections orthographiques de certains éléments, condensation de deux éléments (transfert des notices de l'un vers l'autre), ajout ou suppression d'éléments ¹³;
- le **module d'impression d'index** permet l'impression en listes du contenu des fichiers d'autorité avec, pour chaque élément, le nombre d'occurrences et éventuellement la liste des numéros de référence auxquels ils sont liés. Ceci donne entre autre la possibilité de constituer des catalogues imprimés;
- le **module thesaurus** : cet utilitaire qui n'est pas encore accessible dans la version actuelle permettra de gérer entièrement le thesaurus (établissement des diverses relations entre termes, ...). Ceci ouvrira la possibilité de "l'explosion" de descripteurs en mode recherche (recherche d'un descripteur et de tous ses descripteurs spécifiques) et la recherche assistée par guidage dans le réseau des relations associatives (l'ordinateur propose lui-même de compléter la recherche par l'utilisation de descripteurs sémantiquement proches);
- le **module statistiques** permet d'obtenir une information détaillée sur le contenu actuel de la base de données (nombre et proportion de documents par catégories, par date de publication, par langue, par service, ... proximité des auteurs, fréquence de publication par périodique, par maison d'édition, ... occurrences d'utilisation des mots, des descripteurs, ...);
- le **module réorganisation**. Lorsqu'un certain nombre d'opérations de maintenance ont été menées sur la base de données (modification ou destruction de références, modification des fichiers d'autorité), il est bon de réorganiser les fichiers pour récupérer de la place sur disque et accélérer le travail du logiciel en le débarrassant de tous les éléments devenus inutiles. GABI propose une évaluation préalable de la nécessité d'opérer cette réorganisation. Il s'agit d'une opération lourde pouvant s'avérer extrêmement longue sur des ordinateurs peu puissants et lorsque le nombre

13

Bien entendu, ces deux dernières modifications sont exclues pour les fichiers d'autorités intervenant de manière constitutive dans les références (auteurs, titre de périodique, ...). Seuls peuvent être concernés, le fichier des mots-clefs, l'anti-dictionnaire, la liste des pays.

de références devient élevé (au-delà de 1000 ou 2000). Il est néanmoins indispensable de la réaliser de temps à autre sous peine de voir fléchir la performance du logiciel;

- le **module d'installation**. Il permet d'installer une nouvelle version du logiciel. En fonction du type d'unité de stockage choisi (disque dur ou disquette), plusieurs possibilités sont offertes. Les principales sont une installation "*complète*" qui crée un système totalement opérationnel; ou une version dite "*de transport*" qui crée sur disquette un système de taille réduite ne permettant pas l'introduction ou l'exploitation de références, mais destiné à servir de "vecteur de transport" du logiciel, en vue de l'installation sur un autre ordinateur;
- le **module d'effacement des étapes de recherche préalables** qui permet à l'utilisateur qui souhaite aborder un nouveau thème de recherche "d'oublier" les étapes de recherches antérieures et de revenir à un protocole de recherche vierge. Ce nettoyage permet aussi de récupérer de la mémoire, au cas où GABI aurait averti qu'il arrive aux limites des possibilités de mémoire vive du système.

5. AIDE

Dans le module d'aide, l'utilisateur peut opter pour deux possibilités :

- une **information générale sur la version** utilisée de GABI APA Style : c'est le classique "*A propos de ...*". Celui-ci permet également d'obtenir une indication de la quantité de mémoire vive disponible à ce moment, ainsi que la date de la dernière modification des fichiers;
- l'accès à travers un index à une centaine de **rubriques d'aides** spécifiques qui décrivent pour chaque étape de toutes les fonctions de GABI, les possibilités du système, les touches actives et les réponses ou choix attendus de l'utilisateur.

Ces écrans d'aides sont également accessibles à tout moment de l'utilisation de GABI via l'emploi de la touche < F1 > (voir aide en ligne *CONTEXT SPECIFIC*, page 33).

H. PHASES DE DÉVELOPPEMENT DU LOGICIEL

1. VERSION 1.3

Une première version de GABI a été réalisée dans le courant de l'année 1991. Si les objectifs finaux et la philosophie de l'application étaient déjà définis et l'analyse générale de l'application, largement développée, il s'agissait d'une version limitée n'offrant que les fonctions primitives du logiciel. Seuls les modules **AJOUT DE REFERENCES**, **VISUALISATION**, un début de module de **RECHERCHE** et quelques outils d'**IMPRESSION**, soit le "cœur" du logiciel, étaient fonctionnels. Cette version intermédiaire avait pour objectifs de permettre :

- la saisie de quelques centaines de références des académiques de la FAPSE dans le cadre d'un travail de fin d'études d'une étudiante de l'I.S.I.S. ¹⁴ dont il sera question plus loin (page 56);
- un test grandeur nature de la structure des fichiers et index et du mode d'approche de la référence, afin d'y apporter les corrections nécessaires.

L'absence de module de modification ou de destruction de références (opérations extrêmement complexes et délicates à développer de par la structure relationnelle des fichiers), imposait de faire toute correction manuellement. Cela les rendait inconfortables voire même pénibles.

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 se distingue principalement de la précédente par :

- des modifications dans la structure des fichiers et index;
- l'amélioration de la recherche de références, 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;
- l'ajout des utilitaires;

¹⁴ Institut supérieur d'enseignement des **S**ciences de l'**I**nformation et des **S**ciences sociales de la Province de Liège.

- l'introduction du système d'aide, totalement inexistant dans la première version;
- 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, absents de la première version, a été réalisé. Un des effets les plus visibles de ce travail apparaît dans la structure et le dessin des écrans : multi-fenêtrage, superpositions tri-dimensionnelles et menus déroulants.

3. β TEST

La dernière étape a consisté, d'une part en une optimisation et d'autre part un toilettage final de l'application à travers une phase de β test.

Le β test est une étape extrêmement importante dans le développement d'une application. Sans elle, le développeur risquerait de se complaire dans des stratégies liées à son propre fonctionnement ou au background qu'il possède par rapport à l'application. De plus, un certain nombre de conditions d'erreurs risqueraient de ne pas apparaître car même s'il s'attache à de nombreuses et minutieuses vérifications, le test systématique de tous les cas possibles dans toutes les conditions et pour toutes les fonctions est une tâche irréalisable. Il restera toujours quelques cas particuliers non prévus. En outre, les tests qu'il réalise lui-même ont toujours quelque chose d'artificiel puisqu'ils sont menés en dehors des situations réelles. Il est donc souhaitable de prévoir une phase où un nombre restreint d'utilisateurs pourront manipuler et "torturer" le logiciel à leur guise, en conditions réelles, pour faire apparaître ses défauts et ses limites et permettre d'y remédier.

Un échantillon de personnes différenciées quant à leur compétence informatique, en particulier dans le domaine de l'utilisation des systèmes de gestion des bases de données, ont été confrontées pendant une durée variable à GABI APA Style. Des explications minimales leurs ont été fournies au départ et leurs réactions, stratégies et remarques verbales ont été étudiées.

Ceci a permis d'observer le degré de facilité de compréhension des opérations, de conforter ou infirmer des hypothèses quant à la logique de l'utilisateur, de rectifier ou d'améliorer certaines procédures (par exemple l'installation d'une technique de "trace" pour les combinaisons booléennes permettant à chaque instant à l'utilisateur de savoir où il en est), de mettre en évidence les termes ou opérations nécessitant une explication complémentaire, et enfin, de rencontrer des conditions d'erreur non prévues.

Etant donné le temps imparti, ce β test n'a pu avoir qu'une ampleur limitée (5 personnes). Il est donc certain que bien qu'en nombre probablement limité, un certain nombre de "bugs" et de défauts subsistent dans la version actuelle. Même si une cette phase avait été réalisée à beaucoup plus large échelle, ceux-ci auraient pu être réduits, mais rien ne permet d'affirmer qu'ils auraient disparu. Il en va d'ailleurs ainsi pour toutes les applications développées par des sociétés professionnelles. Le dernier et véritable filtre d'erreurs et source d'amélioration est sans conteste la diffusion réelle de l'application auprès des utilisateurs finaux.

I. DONNÉES TECHNIQUES

1. CONFIGURATIONS HARDWARE ET SOFTWARE NÉCESSAIRES

- GABI APA Style peut fonctionner sur tout ordinateur totalement compatible IBM. Cependant, étant donné l'importance des opérations, un système rapide de type AT 286 ou mieux encore 386 est plus approprié et plus confortable qu'un simple XT.
- Le système d'exploitation doit être au moins égal ou supérieur à MS-DOS 3.3 TM 15.
- La mémoire vive disponible pour l'application doit être au minimum de 445 Ko, mais idéalement, égale ou supérieure à 550 Ko.
- L'utilisation à partir d'une disquette est possible, mais peu confortable, étant donné la fréquence des accès disques. Il est recommandé de l'utiliser à partir d'un disque dur, si possible rapide.
- L'installation automatique peut se faire via n'importe quelle unité de disquette haute densité ou via une unité simple densité 3 pouces et demi (une disquette 360 Ko ne permet pas de recopier le seul fichier principal : GABI.EXE dont la taille dépasse les 635 Ko).
- N'importe quel type d'écran graphique ou texte est suffisant. Cependant, un écran couleur est plus agréable.
- Toute imprimante connectée au système peut convenir, pour autant qu'on dispose ou que l'on crée le pilote d'imprimante correspondant.
- Afin de permettre à GABI d'ouvrir simultanément le nombre de fichiers dont il a besoin, le fichier CONFIG.SYS du système doit absolument comporter la ligne de commande **FILES = 85** et le fichier AUTOEXEC.BAT, la ligne **SET CLIPPER=F80** . Si cela s'avère nécessaire, GABI adapte lui-même ces éléments au moment de l'installation.
- L'utilisation dans le fichier AUTOEXEC.BAT d'une commande telle que **FASTOPEN** munie d'un paramètre d'au moins = **80** est optionnelle mais est bienvenue car elle accélère sensiblement les accès disques répétés. Il en va de même de tous procédés permettant d'accélérer les accès disques (utilisation de mémoire cache, ...).

¹⁵ Les versions de DOS antérieures ne permettent pas d'ouvrir simultanément plus de 20 fichiers.

2. TESTS DE PERFORMANCE

Dans le cadre de l'optimisation de l'application, quelques tests de performance ont été réalisés. Avec toutes les précautions d'usage, ils permettront la comparaison éventuelle avec d'autres applications.

Pour les calculs de vitesse, seules les situations extrêmes ont été mesurées pour donner une idée des limites du système. Dans la plupart des cas habituels, les délais sont incomparablement plus courts, souvent même imperceptibles.

Les tests ont été réalisés sur une base de plus de 1800 références obtenue par incorporation successive des 609 références initiales et ajout de quelques références supplémentaires. Le hardware consiste en un PC 386 cadencé à 25 Mherz, muni d'un disque dur de temps moyen d'accès de 19 msec. Le lecteur adaptera les données en fonction de conditions différentes.

RECHERCHE :

- temps nécessaire pour la recherche par fichiers d'autorité la plus lourde qu'il soit possible d'imaginer : recherche par mots clefs avec troncature illimitée et racine vide (ce qui équivaut à demander de rechercher pour chacun des 2570 mots-clefs, toutes les références qui l'utilisent, puis de combiner avec un "OU" logique tous les résultats partiels) 3' 05 "
- temps nécessaire pour la recherche équivalente sur le champ AUTEUR (258 auteurs différents) 45 "
- temps maximum pour une recherche n'utilisant pas de fichiers d'autorité (LANGUE = FRANCAIS : 1533 références sur 1800) 21 "

AJOUT

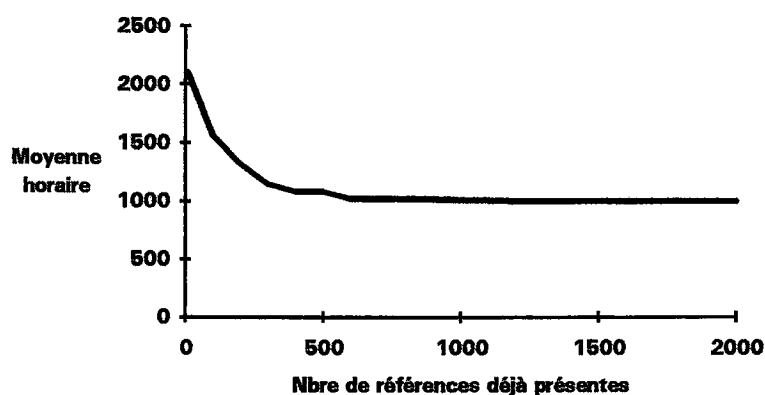
- Délai maximum de vérification pour l'ajout d'une référence (cas d'une référence avec 1 seul auteur qui apparaît dans plus de 350 autres références) 4 "
- Taille totale moyenne occupée par une nouvelle référence (fichiers dictionnaire, fichiers relationnels et index compris) < 5 Ko

IMPORTATION :

Comme sur tout système équivalent, la vitesse d'importation de références en mode ISO est liée au nombre de références déjà présentes dans la base de données. Ceci s'explique par l'importance croissante que prend la mise à jour des index.

Pour GABI, on obtient la courbe suivante pour une importation complète sans aucun doublon et avec une base de données fraîchement réorganisée (les valeurs ont été ramenées en moyenne horaire) :

Vitesse d'importation de références en fonction de la taille de la base de données



Au-delà de 500 références, on observe une quasi stabilisation de la vitesse autour d'une moyenne horaire de 1000 références/heure, soit 17 à la minute.

Ces valeurs correspondent au cas d'importation d'un nombre limité de références (une ou deux centaines) dans une base fraîchement réorganisée. Dans les autres cas, elles peuvent varier très fortement. Ainsi, lorsqu'on introduit de très gros fichiers de plusieurs centaines de références d'affilée la moyenne peut s'affaisser jusqu'à seulement 600 références par heure pour les dernières références. Il faut près de 40 minutes par exemple pour importer un fichier ISO comportant 600 références, ce qui donne une moyenne générale de 900 références à l'heure. Ceci s'explique par le fait que dans ce cas, il n'y a aucune réorganisation intermédiaire des fichiers.

EXPORTATION :

- Vitesse d'exportation en mode ISO, sur disque dur 12600 réf. /heure
- Taille d'un enregistrement ISO 0,5 Ko
- Vitesse d'exportation en mode ASCII, sur disque dur 15000 réf. /heure
- Taille d'un enregistrement ASCII complet avec informations complémentaires 0,44 Ko

CALCUL STATISTIQUE :

- Durée moyenne 1 sec par 50 références

REORGANISATION :

Il s'agit de l'opération la plus lourde du logiciel. Ceci s'explique par le nombre de fichiers à réorganiser (17 bases de données et 45 index dont 9 pour le seul fichier maître) et la taille qu'ils peuvent avoir. Par exemple, le fichier relationnel du dictionnaire compte près de 35000 enregistrements pour une base de données de 1800 références. Ses index dépassent le demi Méga-octet chacun !

- Temps nécessaire pour 1800 références **6 " 15 "**

INSTALLATION :

- Installation complète sur disque dur (sans compter évidemment les délais d'introduction par l'utilisateur du nom générique et du mot de base de la nouvelle base) **29 "**
- Installation d'une version de transport sur disquette (cette valeur est fonction de la rapidité d'accès de l'unité de disquette) **1 ' 20 "**

III. CONSTITUTION DE LA BASE DE DONNÉES DES MEMBRES DE LA FAPSE

Le travail de constitution de la base de données des membres de la FAPSE ¹⁶ s'est fait parallèlement au développement de la première version de GABI APA Style. Pour des raisons de gestion des priorités, il a été mis en arrière-plan au cours de la seconde phase.

Ce travail a été réalisé en collaboration avec Mademoiselle LESENFANTS dans le cadre de son travail de fin d'études en bibliothéconomie à l'I.S.I.S. de 1991.

A. MÉTHODOLOGIE

1. RÉCOLTE DES RÉFÉRENCES

a) PROBLÈME DE SCREENING

Si à priori la récolte des références des publications des membres de la FAPSE paraît une opération extrêmement simple, sa réalisation a immédiatement posé un certain nombre de questions de définitions d'étendue.

- Degré de focalisation inter-individuelle : par membre de la FAPSE, fallait-il entendre toute personne qui à quel titre que ce soit, quelle que soit l'époque et quelle que soit la durée a fréquenté la Faculté ou antérieurement l'Institut, y compris dès lors les chercheurs même de très courte durée, les A.C.S., ... et même les étudiants ? Une telle définition, si d'aventure elle avait été opérationnalisable, aurait réuni un nombre considérable de personnes ! Dès lors, quelle limite fallait-il choisir ?

¹⁶ Dans GABI APA Style, cette base de données porte l'identificateur UDPSY. Par simplicité, il y sera fait référence sous cette dénomination dans le texte ci-dessous.

- **Degré de focalisation intra-individuelle** : pour les membres qui seraient retenus, fallait-il considérer toutes leurs publications, y compris celles réalisées antérieurement à leur entrée dans l'institution, voire même, postérieurement ? ou celles publiées en dehors du cadre universitaire ? Fallait-il inclure les références non publiées au sens strict (communications à des congrès non publiés, notes de cours, rapports à diffusion limitée, mémoire de licence, ...). Fallait-il prendre en considération les traductions d'un même document ?

Aucune décision ne peut être pleinement satisfaisante. Il y aura toujours des cas particuliers qui viendront plaider pour l'un ou l'autre choix.

Dans le cadre du travail de Mademoiselle LESENFANTS qui constitue l'état actuel de la base de données UDPSY, les limites suivantes ont finalement été adoptées : publications scientifiques classiques ¹⁷ en langue originale, des membres académiques actuels de la FAPSE ¹⁸, à l'exclusion des rapports et documents à diffusion privée.

L'objectif final est d'étendre la base de données dans un premier temps à l'ensemble des académiques et scientifiques actuels de la FAPSE, puis ensuite aux académiques ayant quitté le Faculté et enfin, aux scientifiques dans les mêmes conditions, en privilégiant ceux qui sont restés un temps long au sein de l'institution.

b) TECHNIQUE DE RÉCOLTE

L'U.D. ne possédant qu'un nombre très limité de ces documents, 2 courriers successifs ont été adressés aux membres de la FAPSE les sollicitant de transmettre à l'U.D. leur liste de publications, munie si possible des tirés-à-part ou d'exemplaires des documents correspondants.

Malgré de nombreux rappels verbaux, certains académiques ne nous ont pas encore transmis à l'heure actuelle leur liste de publications, par "*manque de temps*". Aussi, la définition de la couverture de la base de données UDPSY a-t-elle du être finalement restreinte aux seuls membres ayant répondu à nos sollicitations (13 académiques sur un peu plus d'une vingtaine, soit 0,59 % seulement). Heureusement, parmi ceux-ci, on compte les académiques les plus prolifiques. Puisque nombre de documents sont écrits en collaboration, ceci renvoie en définitive à 258 auteurs différents.

Chaque fois que possible, nous avons complété les listes par d'autres sources d'informations (documents présents à l'U.D., recoupement avec les listes de publications des scientifiques, ...).

¹⁷ Monographies, ouvrages avec éditeurs scientifiques, chapitres de tels ouvrages, articles de périodiques, communications scientifiques à des congrès et colloques, mémoires de licence et thèses de doctorat, rapports de recherche publiés.

¹⁸ Actuels au moment de la constitution de la base de données, c'est-à-dire pendant l'année académique 1990-1991.

2. ETAT DES RÉFÉRENCES BIBLIOGRAPHIQUES

Des surprises de taille nous attendaient en dépouillant les listes de publications reçues, à l'exception de celles de 3, voire 4 académiques, qui par ailleurs furent les plus rapides à nous transmettre les informations et utilisaient déjà les normes de l'A.P.A.

- la plupart des listes ne suivaient aucune norme. Au mieux, retrouvait-on l'ossature d'une norme personnelle, mais dans ce cas, avec une faible cohérence interne. Des informations secondaires, voire inutiles étaient reprises, mais des éléments essentiels manquaient (les pages, les villes d'édition, voire les dates, ou les collaborateurs, ...). On observait aussi beaucoup de confusions entre les éléments;
- très souvent, elles étaient tirées de curriculum vitae et étaient présentées comme il est malheureusement devenu trop souvent de coutume dans ce type de document : mise en évidence de la personne qui rédige le C.V., avec rejet sous l'appellation trompeuse "en collaboration de" de tous les autres auteurs, même si la personne n'est en fait que le 2^{ème}, voire le n^{ème} auteur effectif du document ¹⁹. Réorganiser les listes d'auteurs par recoupement et vérification n'a pas toujours été une sinécure;
- certaines références présentaient des éléments erronés (ne correspondaient pas au document primaire);
- dans certains cas, les listes présentaient une confusion entre publications scientifiques au sens classique du terme et travaux scientifiques personnels (recherches sans publication, cours oraux, ...);
- quelques listes étaient incomplètes : par recoupement avec d'autres, il a été possible de retrouver des références oubliées par certains auteurs. Rien ne permet donc d'être assuré de l'exhaustivité;
- plus délicat, et sans doute plus grave, chez un auteur, certaines références dites "sous presse" ne correspondaient en fait à aucun document réel. Il s'agissait d'anciennes propositions de textes, ou même d'idée de publications qui n'avaient jamais vu le jour mais qui avaient déjà été ajoutées à la liste de publications du C.V. !

Bien entendu, ces problèmes se rencontraient à des degrés divers selon les auteurs. Néanmoins, il est étonnant et même quelque peu inquiétant de constater un tel état des références pour des académiques. Cela renforce le sentiment de l'urgence de développer un outil tel que GABI APA Style et la base UDPSY.

Chaque fois que possible, une confrontation a été faite entre la référence et le document source et le cas échéant, entre les deux versions d'une même référence rédigées par des collaborateurs. Malheureusement, il n'était pas rare que l'auteur lui-même ne possédât plus aucun exemplaire du document primaire.

¹⁹ Une technique plus saine serait me semble-t-il, de citer dans le C.V. la référence exacte du document. Au lecteur de juger de la part réelle qu'y a prise la personne évaluée.

En cas de doute, l'auteur a été interrogé pour tenter d'obtenir des compléments d'information.

Malgré cela, un certain nombre de références n'ont pu être corrigées de manière suffisante et n'ont dès lors pas été incluses dans la base UDPSY. Elles sont conservées en attente d'informations complémentaires. Les références auxquelles manquaient un seul élément comme la pagination ont été incorporées en remplaçant provisoirement celui-ci par un point d'interrogation.

3. INDEXATION

a) ELABORATION D'UN THESAURUS

Il n'existe pas de thesaurus couvrant simultanément et de manière équilibrée les différents domaines de la FAPSE. Aussi ai-je été amené à en construire un, spécifique à l'U.D.

La première mouture en a été élaborée en collaboration avec une autre étudiante de l'I.S.I.S., Mademoiselle Régine VANDERHEYDEN pour permettre l'indexation thématique des mémoires de licence dans le cadre de son travail de fin d'études en bibliothéconomie (1990). Cette première version a été largement amendée de manière à permettre la couverture de documents de types différents des mémoires de licences que sont les publications scientifiques des académiques.

Le module THESAURUS de GABI n'ayant pas encore été développé, cet outil a été créé au moyen du logiciel de Claude COIBION : G.T.I. (Gestion de Thesaurus Interactive) ²⁰.

Comme tout thesaurus, celui de l'U.D. est inachevé. Il subira encore de nombreuses modifications, et principalement des ajouts, pour s'adapter le plus adéquatement possible à la documentation spécifique de l'U.D. L'objectif qui lui est assigné est en effet beaucoup plus large que l'indexation des publications de la FAPSE présentes dans UDPSY. Il devra permettre l'indexation de l'ensemble des ouvrages de l'Unité de Documentation. Mais ça, c'est une autre histoire ...

CARACTERISTIQUES DU THESAURUS

- Dans sa forme actuelle, le thesaurus compte **783 descripteurs** et **214 termes rejetés**. Il est disponible en version papier à l'U.D.
- Son **taux d'équivalence**, c'est à dire la proportion de non-descripteurs par rapport au nombre de descripteurs est de **0,27**. Plus cette valeur est élevée, meilleure est la cohérence et la précision de l'indexation en raison du plus grand nombre d'entrées alphabétiques du thesaurus. La valeur observée s'approche sans l'atteindre de la valeur idéale qui est de 0,50 (le

²⁰ Ce thesaurus a été développé en 1988, en Clipper et en utilisant comme GABI une technique des fichiers relationnels.

nombre de non-descripteurs égal à la moitié de celui des descripteurs). Il serait donc intéressant d'ajouter encore des équivalences (termes rejetés).

- Son **taux d'enrichissement**, qui indique la richesse des relations hiérarchiques (nombre total de relations hiérarchiques par rapport au nombre de descripteurs) est de **5**. Cette valeur est peut-être légèrement trop élevée. Elle est le signe d'un thesaurus très structuré.
- La **flexibilité** (proportion de descripteurs qui constituent eux mêmes des éléments de descripteurs composés, par rapport au nombre total de descripteurs) est de **0,33**.
- Enfin, le **ratio de précoordination** (qui donne la proportion du nombre de mots significatifs par rapport au nombre total de descripteurs, et par là, est le signe du degré de précision et de discrimination des descripteurs choisis) est de **1,6**. Cette valeur se situe dans la zone considérée comme idéale pour un thesaurus (entre 1,5 et 2 mots significatifs par descripteur).

b) RÉALISATION DE L'INDEXATION

L'indexation des documents introduits dans UDPSY a été faite au moyen de ce thesaurus par Mademoiselle LESENFANTS sous ma supervision.

Dans un certain nombre de cas, l'absence de document primaire ²¹ nous a forcé à choisir les descripteurs sur base du seul titre, ce qui était assurément hasardeux, mais inévitable.

A posteriori, un certain nombre de choix inadéquats apparaissent. Il faut en excuser la jeune étudiante qui se trouvait face à une tâche extrêmement délicate, étant donné son absence de formation scientifique, en particulier dans les disciplines couvertes par notre Faculté. Il est compréhensible dans ces conditions que certaines erreurs se soient glissées. Par ailleurs, il ne m'a pas été possible de revoir les indexations de toutes les notices.

Une procédure systématique devra être mise en place pour corriger ces erreurs. Une telle procédure sera proposée dans les perspectives d'avenir, en page 69.

²¹ En ce qui concerne les références pour lesquelles l'auteur ne nous avait pas transmis de tiré-à-part ou de copie, une recherche a été faite dans les collections de l'U.D. pour tenter d'atteindre le document primaire. Dans un nombre de cas non négligeable, cette recherche s'est cependant révélée infructueuse (titres de périodiques non acquis par l'U.D., ...).

B. CARACTÉRISTIQUES DE LA BASE UDPSY

Au départ, il était extrêmement difficile d'évaluer le nombre de documents qui devraient être introduits dans la base de données. Etant donné les limitations exposées en page 56 et suivantes, le nombre de documents inclus dans UDPSY est actuellement de 609. On peut raisonnablement escompter qu'avec l'introduction des scientifiques et des académiques non encore repris, et en particulier ceux qui ne font plus partie de la FAPSE comme Gilbert DE LANDSHEERE ou François DUYCKAERTS par exemple, on dépassera aisément les 2000 notices.

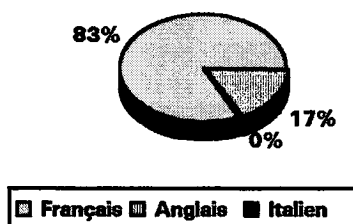
Le lecteur trouvera ci-après quelques caractéristiques de la base actuelle. Elles ont été obtenues pour la plupart directement au moyen du module STATISTIQUE de GABI, sinon en faisant quelques recherches combinées.

Ces chiffres permettent de donner un aperçu des possibilités d'exploitation à des fins descriptives de GABI APA Style. Cependant, il faut les considérer à titre indicatif et avec réserve, car, il est clair qu'on ne peut considérer la base actuelle comme un échantillon représentatif des productions de la FAPSE. Cette photographie permet néanmoins de tirer un certain nombre d'enseignements intéressants.

1. LANGUE ET TYPE DE DOCUMENT

Bien qu'il s'agisse de publications d'académiques, la proportion des documents en français est impressionnante : 82,6 % contre 17,2 % pour l'anglais. Les productions en d'autres langues sont quasi inexistantes (1 seul document en italien).

Part relative des différentes langues dans UDPSY



Les articles de périodiques sont très largement majoritaires : plus des deux tiers des documents (68,2 %). Parmi ceux-ci, 85 % sont en français contre 14 % en anglais.

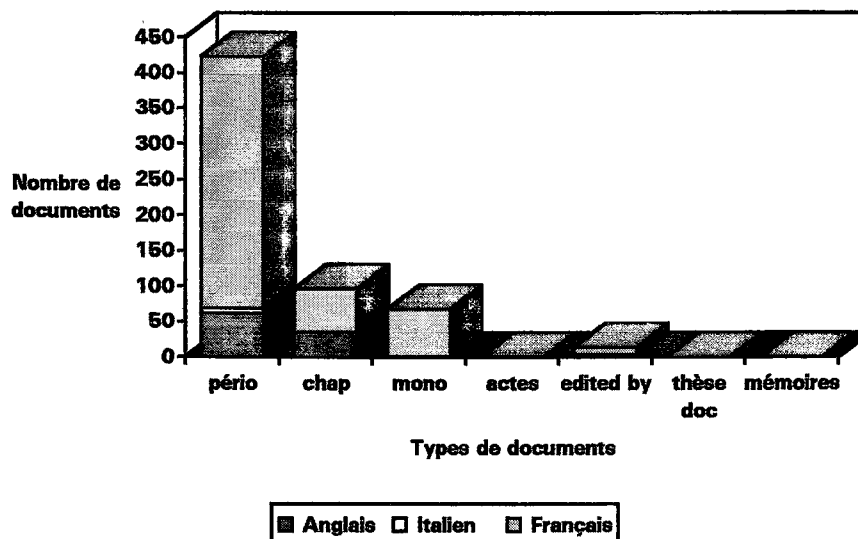
Les chapitres d'ouvrages avec éditeur scientifique suivent de loin et constituent environ un sixième de l'ensemble des documents (15,9 %). La tendance y est différente. La part de l'anglais y est beaucoup plus importante. Elle approche les deux cinquièmes

(36,0 %). Ceci s'explique sans doute par le fait que les ouvrages de ce type qui sont légion chez les anglo-saxons, sont beaucoup moins fréquents en francophonie.

Les **monographies** viennent en troisième position de fréquence (11,1 %). Comme pour les 10 ouvrages "*edited by*" publiés par des membres de la FAPSE, la tendance est très nettement francophone (respectivement 98,5 % et 80 %).

Les 3 types classiques de documents (articles, chapitres d'ouvrages édités et monographies) couvrent 95 % des documents introduits actuellement dans UDPSY. Sans doute y a-t-il là un artefact lié au caractère particulièrement lacunaire et imprécis des références à des rapports de recherche, des actes de congrès non publiés, qui conduit à leur rejet provisoire.

Répartition des documents de UDPSY selon le type et la langue



En ce qui concerne les **articles de périodiques**, 153 titres de revues différents apparaissent, ce qui est le signe de l'extrême diversification de la production de la FAPSE (80 revues, soit plus de la moitié des titres répertoriés, n'ont publié qu'un seul document présent dans UDPSY).

Le périodique le plus prisé est "*Les Feuilles Psychiatriques de Liège*", suivi de près des "*Cahiers de Psychologie Sociale*" et d'"*Education, Tribune Libre*", soit 3 périodiques publiés au sein de notre Université, sinon de la FAPSE. Ensemble cependant, ils ne couvrent que 17,3 % des publications de ce type.

Le premier titre en anglais, le "*Journal of Experimental Analysis of Behavior*", apparaît à la 17^{ème} position au hit parade, avec 4 documents.

L'éditeur liégeois MARDAGA vient largement en tête avec 17,4 % (33 en nombre absolu) des 189 publications de type ouvrage (monographies, ouvrages avec éditeurs scientifiques, chapitres de tels ouvrages, actes de congrès publiés sous forme d'ouvrage avec éditeur scientifique). La première maison d'édition étrangère, les P.U.F arrive en troisième position avec 11 documents. Un très grand nombre de maisons d'édition n'apparaissent qu'une seule fois (66 % des maisons d'éditions représentées).

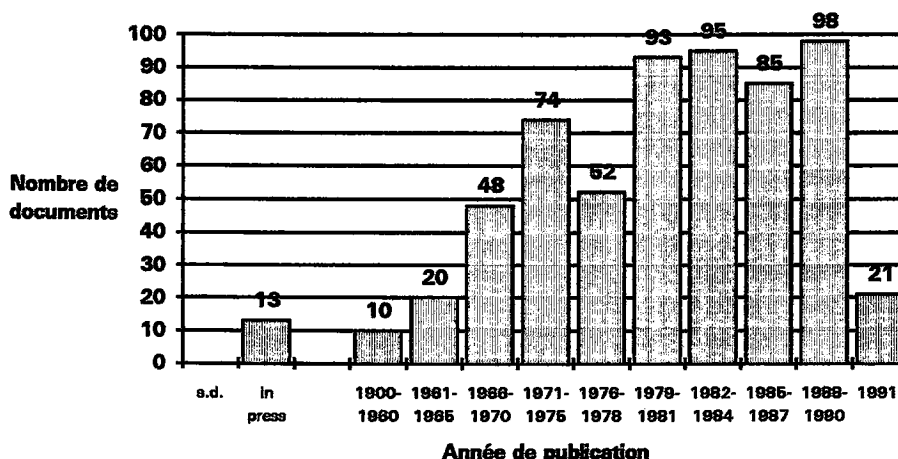
Outre les villes belges (Bruxelles et Liège totalisent ensemble 58,8 % des ouvrages), c'est à Paris (14,9 %) puis à New York (4,8 %) que sont publiés le plus grand nombre de ces documents.

2. DATES DE PUBLICATION

Tous les documents introduits sont postérieurs à 1950. Quasi la moitié (49,0 %) dataient de moins d'une dizaine d'années lors de la phase de récolte (entre 1981 et 1991). Ceci ne peut s'expliquer qu'en partie par "le jeune âge" des académiques de la FAPSE. Une autre hypothèse qu'il faudrait vérifier, est que beaucoup ont perdu ou négligent leurs premières publications. Ceci conforteraient l'idée évoquée en page 4 d'une tendance à la volatilité du patrimoine de publications scientifiques de notre Faculté.

Seuls 2 % des documents étaient "sous presse" (en chiffres absolus : 13). On peut supposer que depuis, la majorité d'entre eux ont été publiés et que dès lors, les références devront être corrigées.

Distribution des documents de UDPSY
selon la date de publication



3. AUTEURS ET SERVICES D'ORIGINE

Parmi les 258 auteurs repris dans UDPSY, on observe 2 locomotives : Marc RICHELLE et Jean Adolphe RONDAL. L'un ou l'autre (parfois l'un et l'autre) apparaît comme auteur dans 40,9 % des références actuellement dans UDPSY (250 sur les 610). De manière très nette, ce sont aussi les auteurs qui publient le plus régulièrement en langue anglaise. Ils assurent à eux seuls les trois quart de ces publications (74,2 %). Il ne reste en fait dans UDPSY que 27 documents en anglais auxquels aucun de ces deux auteurs n'a pris part.

Parallèlement, le même phénomène se reproduit bien évidemment en ce qui concerne les services d'origine des documents : le regroupement des deux services de ces académiques montre qu'ils sont impliqués dans la publication de 46,0 % des documents de UDPSY.

Il sera intéressant d'observer l'évolution de ces données lorsque tous les académiques et les scientifiques auront vu leurs publications incorporées dans la base de données.

4. DESCRIPTEURS ET MOTS-CLEFS

324 des 783 descripteurs actuellement présents dans le thesaurus de l'U.D. ont été utilisés jusqu'à présent dans UDPSY. Etant donné les observations faites à propos des auteurs, il n'est pas étonnant de voir apparaître le plus fréquemment des descripteurs tels que LANGAGE -- DEVELOPPEMENT (9,7 % des documents), COMPORTEMENT ANIMAL, EXPERIMENTATION, CONDITIONNEMENT OPERANT, HANDICAP MENTAL, ENFANT ou PSYCHO-PHARMACOLOGIE, tous thèmes de prédilection des deux auteurs les plus prolifiques.

La fréquence d'apparition du descripteur REVUE CRITIQUE mérite d'être relevée (9,3 % des documents, 2^{ème} descripteur dans l'ordre des fréquences). Elle souligne le renom de nos scientifiques et la confiance qui leur est faite quant à la qualité et l'intérêt de leur avis tant au niveau national qu'international.

Le dictionnaire des mots-clefs compte 2570 termes différents. Les mêmes tendances qu'au niveau des descripteurs s'y observent globalement. Il est intéressant de noter que là on observe l'apparition d'un terme du domaine des sciences de l'éducation, le mot ENSEIGNEMENT, parmi les 10 termes les plus usités.

IV. PERSPECTIVES D'AVENIR

A. AU NIVEAU DU LOGICIEL

La version 2 de GABI APA Style ne constitue pas un aboutissement, mais une simple étape. Il est probable et même souhaitable que le logiciel subira dans l'avenir une évolution constante. D'ores et déjà, avant même que sa diffusion soit réalisée, un certain nombre de possibilités d'amélioration peuvent être envisagées. Si elles n'ont pas été apportées dans la version actuelle, c'est parce qu'elles impliquaient des développements conséquents et qu'il fallait bien s'arrêter de programmer à un moment donné. S'y ajouteront les remarques et propositions des utilisateurs finaux une fois que le logiciel aura été diffusé (par exemple, l'incorporation de cas particuliers des normes A.P.A. non encore gérés par GABI, cfr. page 18).

Quelques unes de ces pistes sont évoquées ci-après. Certaines concernent la structure profonde du logiciel, d'autres sont des améliorations de surface. Aucune n'a encore été soumise à une analyse détaillée pour en évaluer la faisabilité et l'intérêt effectif.

Enfin, il ne faut voir dans leur ordre de présentation aucun échelle de priorité.

1. POSSIBILITÉS D'UTILISER D'AUTRES NORMES

Une amélioration intéressante qui a déjà été évoquée plus haut (page 34) serait de dépasser le cadre des normes de l'A.P.A. et de permettre par un système d'option la sélection de la norme bibliographique que l'on souhaite voir appliquée dans les modules de communication.

Cette idée séduisante de prime abord devrait être étudiée minutieusement dans une double perspective : d'un point de vue technique bien sûr, car il est probable que cette amélioration impliquerait de très nombreux développements, mais aussi d'un point de vue utilisateur : quel serait le rapport coût-bénéfice ? Cette facilité n'impliquerait-elle pas un alourdissement des procédures de saisie de nouvelles références par exemple ?

S'il s'avérait qu'en réalité peu d'utilisateurs sont confrontés à la nécessité de présenter des références selon plusieurs normes différentes, une solution alternative pourrait être de développer des versions parallèles de GABI (ISBD Style, Style Presses Universitaires de France, ...).

2. AJOUT D'UN RÉSUMÉ

Il pourrait être intéressant d'ajouter aux éléments de la référence un résumé du contenu du document. Cependant, cette idée devra également être évaluée selon les deux points de vue évoqués plus haut :

- d'un point de vue technique, est-elle praticable ? Quel serait l'accroissement des fichiers dictionnaires qui en résulterait, et en particulier de ses fichiers relationnels et de leurs index ? Le système resterait-il exploitable sous cette forme et avec un nombre suffisant de documents, sur des ordinateurs de type PC ?
- d'un point de vue utilisateur : il faudra évaluer dans quelle mesure il est réaliste d'imaginer que le personnel de l'U.D. et les membres de la FAPSE auront le temps nécessaire pour composer et introduire des résumés dans GABI. Peut-être l'abstract fréquemment inclus dans les articles de périodiques pourrait-il être utilisé à cet effet.

3. THESAURUS EN LIGNE

Le développement d'un module complet de gestion de thesaurus en ligne, avec entre autre production de listes permutées et hiérarchiques, module prévu initialement, constituerait une amélioration sensible.

- D'une part, il permettrait d'éviter la redondance de l'utilisation d'un outil supplémentaire tel que G.T.I.
- D'autre part, il devrait offrir la possibilité d'exploiter au niveau de la saisie et de la recherche, le réseau de relations arborescentes du thesaurus (possibilités d'extension de la recherche aux termes spécifiques, proposition automatique de termes associés, renvoi transparent des termes exclus vers les termes acceptés, ...)

Il serait intéressant de développer ce module sous une double forme : l'une serait incluse dans GABI et l'autre fonctionnerait sur les mêmes fichiers de descripteurs, mais de manière indépendante. Ceci permettrait l'exploitation d'un thesaurus unique pour diverses applications extérieures à GABI.

4. MODIFICATION NON SÉQUENTIELLE DE RÉFÉRENCES

Actuellement, lorsqu'il souhaite modifier une référence, l'utilisateur est obligé de passer de manière séquentielle, même brièvement, par l'ensemble des étapes de sa création. Dans certains cas, lorsqu'il n'y a qu'une modification mineure à apporter, cela peut s'avérer fastidieux .

Il serait intéressant de permettre à l'utilisateur de choisir en mode pleine page les éléments qu'il souhaite effectivement modifier.

5. RECHERCHE DIFFÉRÉE

En ce qui concerne les procédures de recherche, le choix qui a été fait est en effet celui de la recherche **effective** à chaque étape (repérage de tous les numéros de documents correspondants). Cependant, celle-ci n'est indispensable que dans deux cas : lorsque l'utilisateur désire afficher, imprimer ou exporter les résultats de l'étape de recherche et lorsqu'il souhaite combiner plusieurs étapes. Dans les autres cas, seul le nombre d'occurrences l'intéresse. Or, lorsque le nombre d'occurrences est élevé, le logiciel peut passer un certain temps à identifier tous les documents qui répondent à une équation de recherche.

Les fréquences d'occurrence étant directement accessibles sous GABI, il serait envisageable de modifier les procédures de recherche pour ne réaliser réellement le repérage des documents que lorsque celui-ci est nécessaire. Ces dernières recherches seraient alors ralenties, mais toutes celles où la recherche ne doit pas être effectivement consommée en seraient considérablement accélérées. Ce système aurait comme autre avantage d'économiser la mémoire vive utilisée.

Des essais de condensation de ces étapes de recherches devraient cependant être menés au préalable, afin d'en évaluer le bénéfice réel. Il faudrait en particulier vérifier si la combinaison booléenne d'étapes de recherche à nombre élevé d'occurrences ne serait pas alors d'une lenteur inacceptable (concentration de procédures longues).

6. SUPPRESSION DU RAPPEL INTERMÉDIAIRE DES ÉTAPES LORS DE LA RECHERCHE PAR ÉTAPES

Lors de la récupération d'une recherche préalable (recherche par étape) pour une combinaison booléenne, GABI reprend en une étape intermédiaire la recherche rappelée. Cette redondance serait avantageusement éliminée. Cela améliorerait la lisibilité du protocole de recherche et allégerait d'autant la charge en mémoire vive. Cette évolution mineure en apparence implique cependant des modifications relativement importantes dans les modules de recherche.

7. POSSIBILITÉS DE COMBINAISONS SUPPLÉMENTAIRES

Les opérateurs booléens sont les outils de combinaisons les plus utiles. Cependant, dans la recherche par mots-clefs, il serait intéressant d'offrir également d'autres possibilités de combinaisons. Ainsi par exemple la recherche par **co-occurrences**, ou même des **opérateurs de proximité** pourraient s'avérer très utiles ²².

22

Les opérateurs de co-occurrence permettent de rechercher deux termes présents simultanément dans la même phrase, ici 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 au plus, ou à exactement, X mot(s) d'un second. Il peut même être possible de spécifier l'ordre dans lequel ces deux termes doivent apparaître.

8. UTILISATION DE LA SOURIS

Etant donné sa commodité d'utilisation, la souris est devenue un périphérique de plus en plus largement utilisé. La plupart des machines actuelles en sont désormais équipées et beaucoup d'utilisateurs Macintosh ou WINDOWS [™] sont affolés lorsqu'il faut plus qu'un "simple clic" pour utiliser un logiciel ...

Une des prochaines améliorations de GABI sera par conséquent d'en permettre l'utilisation, afin de simplifier la navigation.

9. ADAPTATION AU MACINTOSH

Etant donné l'omniprésence des Macintosh dans les secrétariats de la FAPSE, il faudra envisager les possibilités de transfert de l'application sur ce système, si l'on souhaite qu'il soit largement utilisé. Cependant,

- il semble qu'il n'existe pas d'utilitaire de traduction directe du code machine d'un système à l'autre;
- il n'existe pas de version Macintosh du langage Clipper. Il n'est donc pas possible de recompiler le programme directement dans cet environnement.

Pour éviter de devoir réécrire l'ensemble du programme dans un langage disponible sur Macintosh (!), la seule possibilité qui semble exister serait alors d'utiliser certains utilitaires du monde Macintosh, qui permettent d'ouvrir sur ces machines une fenêtre MS-DOS et d'y faire tourner directement une application écrite pour fonctionner sur un PC. Il faudra évaluer le degré de compatibilité et tester les conséquences sur la performance de l'ajout de cette couche intermédiaire supplémentaire. Enfin, il n'est pas sûr que cet utilitaire puisse fonctionner sur tous les types de Macintosh.

10. DÉVELOPPEMENT DE DRIVERS PARTICULIERS

GABI permet d'ores et déjà le développement de drivers d'imprimantes spécifiques afin d'exploiter les possibilités de polices de caractères et des attributs de caractères des imprimantes utilisées.

Il serait intéressant d'essayer de développer un driver de ce type pour le mode postscript, largement répandu dans le monde des imprimantes lasers. J'avoue cependant n'avoir aucune idée actuellement du degré de difficulté, voire de faisabilité d'une telle entreprise.

Dans la même ligne, une des perspectives 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 déjà, 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.

B. EN CE QUI CONCERNE LA BASE DE DONNÉES

1. STRATÉGIE DE CORRECTION

Jusqu'à présent, seules des corrections formelles ont été apportées aux références de la base UDPSY. Pour régler les quelques problèmes concernant l'indexation, il me semble intéressant de mettre en oeuvre une procédure systématique et décentralisée.

En effet, il serait sans doute possible de m'atteler seul à la vérification de toutes les indexations réalisées. Mais, si un certain nombre d'erreurs pourraient être ainsi corrigées, il n'est pas du tout certain qu'il me serait possible de réaliser une indexation adéquate dans tous les cas. Il est difficile d'être un spécialiste éclectique (!), Ceci est particulièrement aigu dans le cas des documents dont l'U.D. ne possède pas de copie.

Par contre, nous avons la chance d'avoir les auteurs à portée de main, ce qui est rare en bibliothéconomie, et c'est sans conteste eux qui sont le plus au fait du contenu de leurs propres publications. Par conséquent, il me semble intéressant de leur adresser une liste de leurs publications en leur demandant de contrôler les références et de donner en particulier leur avis sur l'indexation. Il suffirait alors de traduire leurs remarques dans le langage documentaire du thesaurus. Ceci permettrait en outre d'autres corrections plus ponctuelles.

Une fois la base corrigée, la procédure pourrait être reprise de manière régulière pour les mises à jour.

Cette stratégie aurait aussi l'avantage de présenter une production issue de GABI et partant, peut-être de susciter la curiosité et le désir d'utilisation.

2. STRATÉGIE D'ACCROISSEMENT

Quelle que soit sa qualité et son intérêt, l'alimentation d'une base de données pose toujours de sérieux problèmes en termes de personnel. Ce seul facteur vient souvent ruiner tous les efforts de développement d'une base de données. Rares sont les cas où l'on accepte de dégager le personnel nécessaire et d'en supporter la charge. En ce qui concerne les bases de données développées par des bibliothèques, il est difficile de demander aux responsables d'assurer encore cette surcharge.

Cependant, si les membres de la FAPSE peuvent être convaincus par le logiciel, on pourrait envisager une stratégie d'accroissement décentralisé avec bénéfices réciproques, grâce à la potentialité d'exportation/importation en mode ISO qu'offre GABI.

Il suffirait que chaque auteur de la FAPSE, ou chaque secrétariat, possède une version du logiciel et constitue sa base de données personnelle. D'une part, il y introduirait, ses propres publications au fur et à mesure de leur parution, ce qui lui permettrait de tenir aisément à jour sa liste de publications personnelles dans le cadre de son C.V. par

exemple. D'autre part, il pourrait y ajouter toutes les références dont il a besoin pour des publications, demandes de crédits de recherche ..., les sélectionner, imprimer des listes sélectives, en supprimer, ...

La dernière étape serait alors, de manière régulière, de sélectionner les références personnelles postérieures à la dernière mise à jour de la base principale UDPSY localisée à l'U.D., de les exporter en mode ISO sur disquette, de les incorporer ensuite dans UDPSY puis de les modifier éventuellement et de leur ajouter des descripteurs. Ceci réduirait largement la part de travail à assumer par les gestionnaires de la base de données.

Cette organisation idéale serait profitable à chaque partenaire. Elle permettrait la mise à jour régulière et à coût en personnel réduit de cette base de données des publications des membres de la FAPSE.

Bien sûr, on me qualifiera d'idéaliste et de rêveur. J'entends déjà poindre des réticences du style *"Oui, mais nous on préfère les Macs ..."*

3. CRÉATION DU FONDS DE DOCUMENTS

Enfin, afin d'atteindre le deuxième objectif principal de conservation du patrimoine intellectuel de la Faculté, il faudra mettre rapidement sur pied le fonds de documents primaires au départ de ceux récoltés pour servir à l'indexation. Une cote individuelle reprise dans GABI leur sera attribuée pour permettre une localisation aisée.

Il faudra définir une politique pour récupérer des exemplaires des documents manquants, en particulier ceux perdus même par leurs auteurs (prêt interbibliothèques, ...) et dégager les moyens correspondants.

Enfin, le Conseil de Faculté devrait se prononcer pour exiger des membres de la FAPSE que désormais un exemplaire de chacune de leurs publications soit systématiquement déposé à l'U.D. pour être incorporé à ce fonds et à UDPSY. Sans une telle politique de **dépôt obligatoire**, suivie de manière rigoureuse, il y a fort à parier que les habitudes individualistes anciennes reprendront rapidement le dessus et que patrimoine intellectuel de notre Faculté continuera à s'évanouir comme sable serré au creux de la main. Espérons qu'il n'en sera pas ainsi et que le présent travail n'aura pas été vain.

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.
- 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 - 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.

TABLE DES MATIERES

Avertissement.....	2
I. Introduction : motivation du développement de GABI APA Style	4
A. Un constat : la situation des publications des membres de la FAPSE	4
1. Souhait du Conseil de Faculté	4
2. L'Unité de Documentation	5
B. GABI et la constitution d'une base de données des publications de la FAPSE	6
II. Développement du logiciel	8
A. Choix initiaux	9
1. Choix de la norme bibliographique de l'A.P.A.	9
a) Souci d'efficacité et de sobriété.....	10
b) Description succincte	10
c) Limites.....	11
2. Nécessité de développement d'une application spécifique	11
a) Inadéquation de Liber © dans ce contexte particulier	11
b) Absence de produit permettant de gérer des bases de données avec les normes de l'A.P.A.	12
3. Choix de Clipper TM	14
a) Caractéristiques et avantages.....	14
b) Principales limites	15
c) Choix de la version	16
B. Eléments de la référence	18
1. Aménagements par rapport aux normes de l'A.P.A.	18
a) Gestion des cas particuliers et exceptions	18
b) Autres aménagements.....	19
2. Informations complémentaires.....	19
C. Structure des fichiers et index	21
1. Bases de données relationnelles	21
a) Principe.....	21
(1) LES BASES DE DONNÉES CLASSIQUES	21
(2) LES BASES DE DONNÉES RELATIONNELLES	23
b) Application à GABI	24
2. Champs.....	24
a) Champs à longueur fixe et à longueur variable	24
b) Accepter une certaine redondance	25
3. Index préétablis	26
4. Fichier des mots-clefs et anti-dictionnaire multi-lingue	27
D. Convivialité et ergonomie logicielle	30
1. Niveau superficiel.....	30
a) Gestion des écrans	30
b) Assistance à la saisie	31
(1) Simplification de la saisie	31
(2) Utilisation des listes d'autorité	31
(3) Auto-corrections	32

(4) Vérification de la cohérence interne	32
c) Aide en ligne "context specific"	33
d) Recherches pré et post-coordonnées	33
e) Paramétrisations	33
2. Niveau profond	34
a) Logique de l'utilisateur	34
b) Procédures de vérification des doublons	36
c) Enregistrement différé	36
d) Souplesse	37
e) Contrôle permanent de l'état du système	37
E. Gestion de la mémoire	38
1. Rationalisation des lignes du code	38
2. Structuration en overlays	39
a) Principe	39
b) Overlays statiques, overlays dynamiques	39
c) Application à GABI : 3 séries et 2 niveaux	40
3. Sauvegardes temporaires	42
F. Ouverture du logiciel	43
1. Exportation en format texte	43
2. Fomat d'échange ISO 2709	44
G. Description des principales fonctionnalités du logiciel	46
1. Recherche	46
2. Gestion	47
3. Communication	47
4. Outils	48
5. Aide	49
H. Phases de développement du logiciel	50
1. Version 1.3	50
2. Version 2.00	50
3. β test	51
I. Données techniques	52
1. Configurations hardware et software nécessaires	52
2. Tests de performance	53
III. Constitution de la base de données des membres de la FAPSE	56
A. Méthodologie	56
1. Récolte des références	56
a) Problème de screening	56
b) Technique de récolte	57
2. Etat des références bibliographiques	58
3. Indexation	59
a) Elaboration d'un thesaurus	59
b) Réalisation de l'indexation	60
B. Caractéristiques de la base UDPSY	61
1. Langue et type de document	61
2. Dates de publication	63
3. Auteurs et services d'origine	64
4. Descripteurs et mots-clefs	64
IV. Perspectives d'avenir	65
A. Au niveau du logiciel	65
1. Possibilités d'utiliser d'autres normes	65
2. Ajout d'un résumé	66
3. Thesaurus en ligne	66
4. Modification non séquentielle de références	66
5. Recherche différée	67

6.	Suppression du rappel intermédiaire des étapes lors de la recherche par étapes	67
7.	Possibilités de combinaisons supplémentaires.....	67
8.	Utilisation de la souris.....	68
9.	Adaptation au Macintosh	68
10.	Développement de drivers particuliers	68
B.	En ce qui concerne la base de données	69
1.	Stratégie de correction.....	69
2.	Stratégie d'accroissement	69
3.	Création du fonds de documents	70
BIBLIOGRAPHIE.....		71
TABLE DES MATIERES.....		72