GESTION D'UNE BASE DE CONNAISSANCES SOUS TURBO-PROLOG

Ce logiciel répond à une question pratique, que tout étudiant ou tout chercheur s'est un jour posée : comment gérer rationnellement un grand nombre de notes de lecture ? Ce n'est pas une surprise, aujourd'hui, Turbo-Prolog offre tous les moyens de résoudre efficacement ce problème.

En général, pour prendre des notes de lecture, garder dans un coin des idées griffonnées en quelques lignes sur un morceau de papier, ou plus généralement stocker un ensemble de documents écrits fragmentaires, on s'en remet à la bonne vieille technique des fiches cartonnées. Même si cette dernière a depuis longtemps fait ses preuves, on en connaît aussi les limites : comment classer ces fiches ?

On peut imaginer des dizaines de classifications différentes, et le plus souvent, à chaque recherche, on est quitte pour un rebalayage systématique de tout le fichier.

Ce logiciel se propose de résoudre le problème en tirant parti des possibilités de Turbo-Prolog tant au niveau du traitement de l'information (manipulation de listes, base de données dynamique), que de sa restitution (utilisation de fenêtres, de menus, de l'éditeur interne, etc.).

Eviter les mauvaises notes...

Par définition, des notes de lecture, ou des notes personnelles, sont des informations imprévisibles, difficilement normalisables. On ne peut facilement les ordonner en base de données traditionnelle, comme on le ferait avec un simple fichier d'adresses comportant des rubriques fixes « nom », « prénom », « nom de rue », etc. Tout au plus, dans une telle base de données, on ne pourrait qu'indexer les fichiers « texte » par un nombre fini de mots-clefs, nombre qu'il faudrait impérativement préciser lors de la conception de la base. Cela peut paraître limitatif lorsqu'on ne sait pas, *a priori*, combien de mots-clefs seront nécessaires pour référencer un document.

L'emploi, sous Turbo-Prolog, des clauses dynamiques en database, ainsi que des listes, lève la difficulté. Dans le logiciel, deux clauses sont utilisées pour construire la BDD. La première, doc(), contient les informations permettant d'identifier l'ouvrage duquel a été tirée la note de lecture : son titre, son auteur, etc., ainsi qu'une première liste de mots-clefs, et un numéro de référence interne attribué automatiquement par le logiciel.

La seconde clause, tex(), définit la note de lecture proprement dite, à savoir : l'ouvrage de référence, une liste de mots clés identifiant son contenu, ainsi que le nom du fichier en mémoire de masse dans lequel se trouve le texte *in extenso* (cf. figure 1). On dispose ainsi, en mémoire vive, d'une BDD bibliographique couplée à un système d'indexation par mots-clefs de fichiers « texte » résidant sur disque dur ou sur disquette.

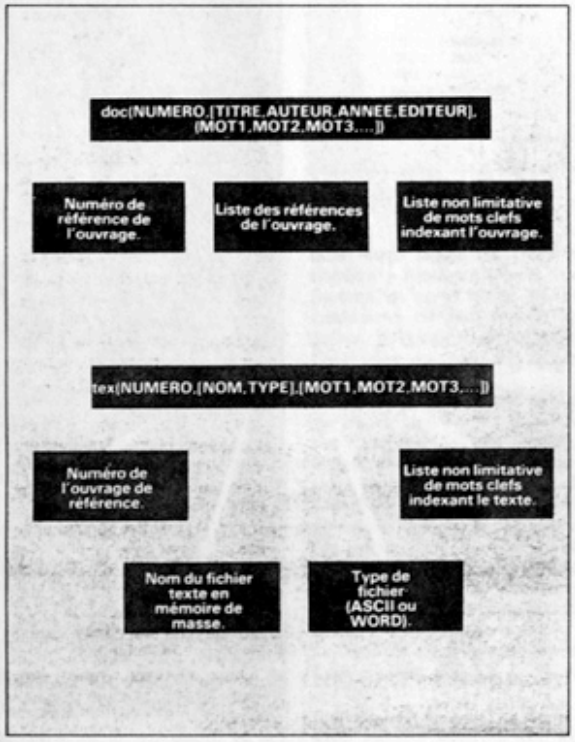

Fig. 1. – Description du fonctionnement des clauses dynamiques doc() et tex().

Création de la BDD

Mais revenons à l'aspect pratique du programme.

Au démarrage, l'écran est

découpé en quatre fenêtres, nommées respectivement « MENU », « MOTS-CLEFS », « REFERENCES », et « FICHIER ACTIF ».

Dans la fenêtre « MENU », il sera d'abord demandé si l'on désire travailler sur une ancienne BDD, ou en créer une nouvelle.

En choisissant la première option, on voit apparaître à l'écran la liste des fichiers possédant une extension « .BDD ». Il suffit alors de pointer avec les touches « curseur » le fichier désiré. Au niveau de la programmation, cette opération est effectuée grâce à une clause Turbo-Prolog très utile :
dir("C:/DOCU","*.BDD",NN)
le premier paramètre étant le nom du directory (on a supposé qu'il s'agissait du sous-directory « DOCU » sur disque dur ; pour un PC double drive, il faut donc rectifier par «A:/DOCU »), le deuxième un masque de sélection (tous les fichiers auront l'extension «.BDD»), et enfin le dernier un paramètre en output correspondant au choix de l'utilisateur.

Si au contraire on désire créer une nouvelle base de données, il suffira de donner son nom, et le système complétera de lui-même en créant un fichier avec l'extension «.BDD» dans le sous-directory « DOCU » (que l'on aura, bien entendu, préalablement créé).

Il ne reste plus alors qu'à rentrer ses informations en jonglant avec les menus, de fenêtre en fenêtre !

Sans rentrer dans le détail de toutes les opérations possibles (résumées en **figure 2**), précisons la marche à suivre pour créer une première base de données.

Le problème est simple : on a sous la main un ensemble de notes tirées de différents ouvrages. On commencera donc logiquement par enregistrer les références de ces ouvrages, en « cliquant » sur « Références Bibliographiques » dans le menu principal (cf. **figure 3**). Apparaîtra alors un menu plus détaillé dans lequel on choisira l'option « Ajouter Référence » (notons qu'après toute fausse manœuvre, on peut revenir au menu précédent et effacer l'opération en cours en tapant sur « escape »).

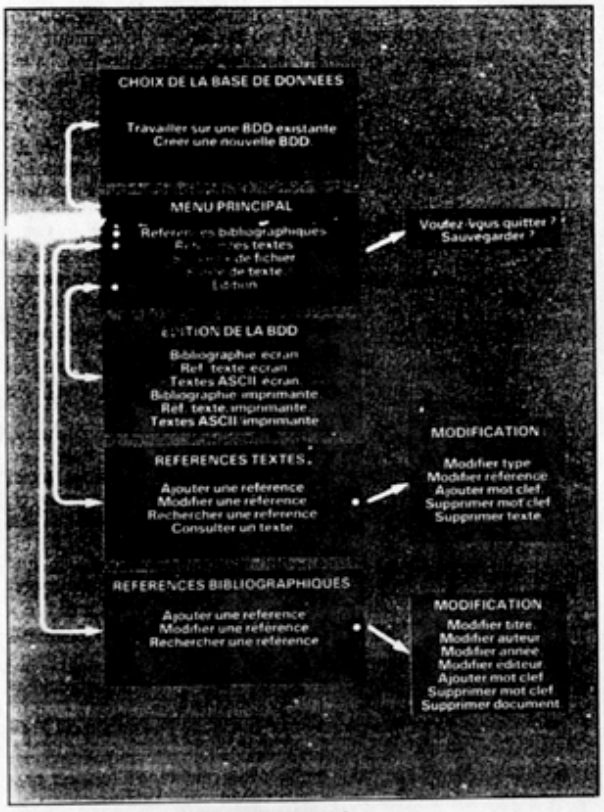

Fig. 2. – Description fonctionnelle du logiciel.

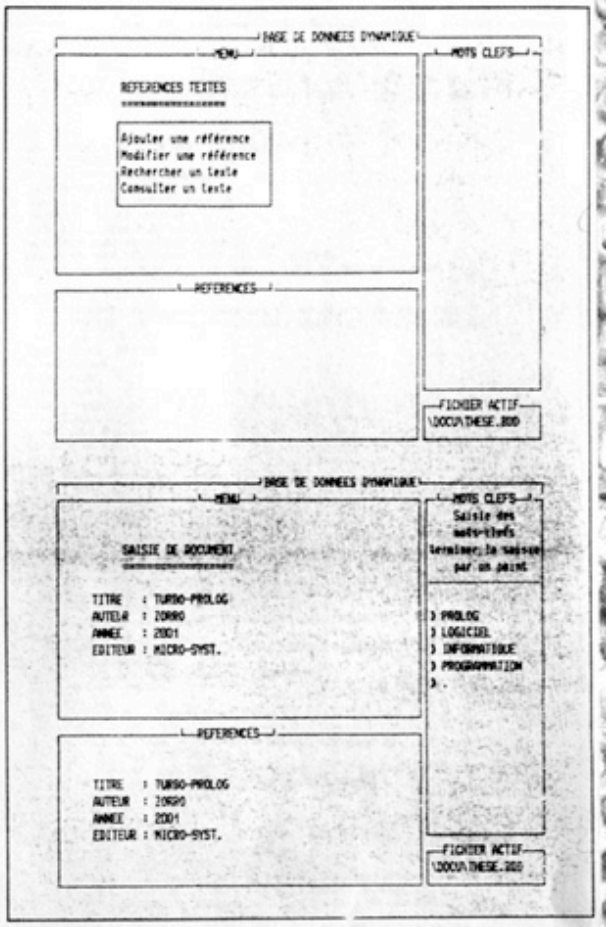

Fig. 3. – Exemples d'écran.

Dans le premier écran de saisie seront rentrés les paramètres classiques : Titre, Auteur, Année, et Editeur. Puis le curseur passera à la fenêtre « MOTS-CLEFS » où sera enregistrée la liste des mots, terminée par un point. Comme on l'a déjà précisé, les mots-clefs sont stockés sous forme de liste. Leur nombre n'est donc pas limité.

Dans le programme, cette saisie s'opère grâce à la clause récursive :
motclef(L, MM),
dans laquelle L est la liste actuelle mémorisée (à l'appel, L=[]), et MM la liste finale.

Si, au terme de cette saisie, on constate une erreur sur l'un des paramètres, on choisira l'option « Modifier Référence » dans le menu « Références Bibliographiques ». Le document sera retrouvé en donnant son titre, ou du moins les premières lettres du titre le définissant sans ambiguïté. Des modifications peuvent être apportées aux paramètres ; il sera également possible d'ajouter ou d'effacer des mots-clefs. Une fois la bibliographie enregistrée, on rentre les textes en cliquant sur « Saisie Texte » dans le menu principal.

Ici est utilisée une nouvelle ressource de Turbo-Prolog : son éditeur interne. En effet, grâce à la clause :

edit(string,string),
on obtient en retour une chaîne de caractères rentrée en écran pleine page (la validation du texte étant effectuée par la touche F10).

Bien entendu, cet éditeur, le même que celui utilisé lors de la programmation, n'a pas la qualité d'un logiciel de traitement de texte. Mais il est suffisant, nous semble-t-il, pour saisir les petits textes que sont des notes de lecture. Pour les puristes, le recours à un logiciel plus performant tel que Word n'est pas à exclure. Comme nous le verrons plus loin, cette éventualité est prévue, mais risque de poser quelques petits problèmes lors de l'affichage du texte à l'intérieur du programme (petits problèmes qui, n'en doutons pas, pourraient facilement être résolus par tout programmeur prenant le temps de se pencher dessus).

La dernière étape de la saisie, enfin, consiste à indexer les textes en choisissant les options « Références Textes » dans le menu principal, puis « Ajouter Référence » dans le menu secondaire. On donne alors le numéro de l'ouvrage de référence – numéro attribué par le logiciel, et précisé lors de la saisie des références bibliographiques. S'affiche en écho, dans la fenêtre « Références », l'intitulé complet de l'ouvrage, puis est demandé le nom du fichier texte, et son type de codification : ASCII s'il a été enregistré grâce à l'éditeur Turbo-Prolog, Word s'il a été écrit sous Word. La BDD est terminée, et prête à l'emploi. Grâce à l'option « EDITION », on peut la lister partiellement ou complètement sur l'écran ou l'imprimante.

On commence alors à profiter de la rapidité et de la souplesse de Turbo-Prolog en recherchant des références (textes ou ouvrages) sur leurs mots-clefs.

Avec l'option « Rechercher une référence » dans le menu « Références Textes », par exemple, le programme listera tous les textes indexés par une sous-liste donnée de mots-clefs, grâce à la clause récursive « rechtexte ».

Extensions du logiciel

Ce logiciel permet de créer une base de données relationnelle en profitant d'un grand nombre de « plus » propres au langage Turbo-Prolog. Il serait néanmoins dommage de s'arrêter en si bon chemin.

La BDD étant entièrement structurée par des « clauses » et des « listes », on peut, et c'est à la vérité un de nos objectifs en utilisant ce programme, la réemployer dans d'autres logiciels, par exemple dans un système expert.

Ces techniques permettront, en particulier, de répondre à deux types de problèmes : comment classer les notes de lecture et les regrouper en ensembles homogènes d'une part, et comment faciliter le processus de recherche en élargissant de façon logique ces recherches d'autre part. Moyennant un bon système de déduction entre mots-clefs, la définition de liens entre synonymes, etc., on peut, à terme, imaginer instaurer un semblant de dialogue entre la BDD et son utilisateur.

Même modeste, un tel logiciel serait de première utilité pour un chercheur.

O. Boisard

```
/*                  *** BASE DE DONNEES RELATIONNELLE ***                  */
/*                                                                         */
/*                    COPYRIGHT OLIVIER BOISARD  87                        */

code=2500
trail=300

domains
   liste=symbol*
   file=destination

database
   doc(integer,liste,liste)
   tex(integer,liste,liste)


/*                        *** EDITION DES MENUS ***                        */
/*                                                                         */
/*   Ce module permet d'afficher des menus à choix multiple.               */
/*                                                                         */
/*                        *************************                        */
```

```
/*                  *** DEFINITION DES PREDICATS ***                       */
/*                                                                         */
/*  Dans cette section sont regroupés tous les prédicats du programme      */
/*                                                                         */
/*                        *******************************                  */

domains
   loc=integer*

predicates
   nondeterm nominal(symbol,symbol)
   nondeterm baratin(symbol,symbol)
   nondeterm document
   nondeterm texte
   nondeterm bunthibli
   nondeterm buntteste
   nondeterm modtexte
   nondeterm modbibli
   nondeterm orientation(integer,symbol)

goal
   makewindow(3,104,6,"F, REFERENCES ,C,,,",15,0,10,60),
   makewindow(4,97,6,"FICHIER ACTIF",22,60,3,20),
   shiftwindow(2),shiftwindow(3),
   shiftwindow(4),shiftwindow(1).

/*                        *** MENU PRINCIPAL ***                           */

debut_du_programme if
   ecran,
   shiftwindow(1),
   nl,nl,
   write("          CHOIX DE LA BASE DE DONNEES"),nl,
   write("          ---------------------------"),
   menu(6,10,["Travailler sur une Base de Données existante",
   clear_memory
   repeat
   ecran
   debut_du_programme
```