Document généré le 9 mai 2025 11:11

Documentation et bibliothèques

LA MICRO-INFORMATIQUE AUTOMATIQUE

Gérard Mercure

Volume 32, numéro 4, octobre–décembre 1986

URI : https://id.erudit.org/iderudit/1052616ar
DOI : https://doi.org/10.7202/1052616ar

Aller au sommaire du numéro

Éditeur(s)

Association pour l'avancement des sciences et des techniques de la documentation (ASTED)

ISSN

0315-2340 (imprimé)
2291-8949 (numérique)

Découvrir la revue

Citer cet article

Mercure, G. (1986). LA MICRO-INFORMATIQUE AUTOMATIQUE. *Documentation et bibliothèques*, *32*(4), 135–143. https://doi.org/10.7202/1052616ar

Tous droits réservés © Association pour l'avancement des sciences et des techniques de la documentation (ASTED), 1986

Ce document est protégé par la loi sur le droit d'auteur. L'utilisation des services d'Érudit (y compris la reproduction) est assujettie à sa politique d'utilisation que vous pouvez consulter en ligne.

https://apropos.erudit.org/fr/usagers/politique-dutilisation/

Cet article est diffusé et préservé par Érudit.

Érudit est un consortium interuniversitaire sans but lucratif composé de l'Université de Montréal, l'Université Laval et l'Université du Québec à Montréal. Il a pour mission la promotion et la valorisation de la recherche.

https://www.erudit.org/fr/

LA MICRO-INFORMATIQUE AUTOMATIQUE

Pour les pionniers de la micro-informatique, les ordinateurs sont d'abord des jeux de meccano à démonter et des programmes à décripter avant d'être des outils de travail à exploiter. Une nouvelle génération d'utilisateurs jouit du privilège de pouvoir sauter cette étape du patient bricolage pour entrer de plein pied dans l'ère de l'icône et de l'application immédiate. Sans même devoir apprendre à programmer, ou presque...

Il n'est plus nécessaire d'investir autant de temps dans l'apprentissage de base. Lancer un programme ne requiert plus un entraînement d'astronaute. Car les logiciels cachent de plus en plus sous le capot les clés d'ajustement qui n'ont plus de raison d'apparaître au tableau de bord. Les paramètres sont fixés implicitement, «par défaut». Les environnements graphiques à la Macintosh simplifient les commandes du système d'exploitation. Même la chaîne d'initialisation de l'imprimante, série de codes en hexadécimal qui commande le choix de caractères de l'imprimante, est maintenant enrobée du nom de l'imprimante-type. Encore faut-il que l'installation du matériel et du logiciel soit déjà faite pour qualifier d'abord facile cette nouvelle approche de la micro-informatique... Pour la rendre d'un commerce encore plus agréable, on l'a tout récemment dressée à la programmation automatique et même dotée d'intelligence artificielle.

Programmation automatique et intelligence artificielle

Après la conquête facile du traitement de texte et un peu plus ardue de la feuille électronique, on s'attaque maintenant à la base de données relationnelle plus puissante et plus prometteuse, mais moins conviviale. Celle-ci renvoie à des concepts plus difficiles à saisir et exige la connaissance d'un langage de programmation particulier. Aussi met-elle du temps à s'imposer dans l'entreprise. Les menus, les icônes et les écrans d'aide n'ont pas jusqu'à ce jour réussi à en déjouer la complexité. Elle est, comme la fusion atomique, pleine de promesses mais difficile à harnacher.

Les concepteurs se tournent maintenant vers l'automatisme des générateurs de programmes et l'intelligence artificielle (IA) pour rendre les systèmes de gestion de bases de données (SGBD) d'utilisation plus facile. Ils ont réussi à rapprocher la procédure d'interrogation du langage naturel. Le premier logiciel à utiliser ainsi une syntaxe près de la langue courante fut Clout, un module de R:base 4000, une base de données relationnelle. Q & A, une autre base de données, mais non relationnelle celle-là, fait appel au langage naturel non seulement comme mode de repérage, mais aussi pour modifier les données sous forme

d'instructions se rapprochant du langage écrit. Hal de Lotus, constitue la dernière innovation dans le champ cette fois de la feuille électronique. Hal interprète les commandes écrites selon une syntaxe simple comme «Sum Col E» ou «Graph Table», soit pour faire un calcul ou pour tracer un graphique. Paradox, de la famille des bases relationnelles, a aussi recours à des techniques d'IA, mais plutôt pour faciliter la création des relations entre fichiers et rubriques que pour simplifier l'interrogation.

Paradox et son progiciel générateur d'application ApGen

Avec Paradox, les liens entre fichiers s'établissent grâce à des «exemples» (ici, BIBLIO) apportés sous la rubrique commune aux fichiers que l'on veut associer. Un système expert incorporé à la base fait le reste, soit d'écrire les enchaînements et de choisir le chemin le plus court pour arriver à la réponse en créant automatiquement les index nécessaires. Paradox a recours comme mode d'interrogation à une sorte d'illustration du fichier interrogé, ce qui ajoute un élément visuel de représentation. Il s'agit alors de cocher simplement la rubrique désirée ou d'inscrire le descripteur sur lequel porte la question, ou d'introduire les opérateurs dans les cases appropriées. (Fig.1)

L'utilisateur peut sauvegarder le scénario de sa question pour une interrogation future. Veut-il l'enchaîner à un autre scénario? Rien de plus simple! Comme s'il s'agissait d'un magnétoscope, il part l'enregistrement qu'il coupe une fois la scène jouée. Pour reconstituer la scène affichée sur l'écran de l'exemple ci-haut, il suffit d'appeler le script «Fig.2» qui à son tour exécute Figure1. Il est même possible de visionner ou lire l'enchaînement des instructions qui ont permis ce retour à l'écran de cette image.

Le metteur en scène peut modifier directement ce scénario, en revisant le texte ou en faire la reprise sur le plateau aussi longtemps qu'il n'est pas satisfait du jeu. Il réalise aussi les éléments d'un programme dont il fait ensuite le montage. Ces scénarios que Paradox appelle «scripts» sont en fait des macro-instructions, une forme déguisée de programmation, il faut bien le dire, mais qui dispense de l'apprentissage d'un langage formel. La table de montage, c'est le générateur d'application «ApGen» qui permet de créer un logiciel d'auteur, sans écrire une ligne de programmation. ApGen produit automatiquement le programme pour autant que l'opérateur suive docilement un menu qui à son tour crée un autre menu auquel s'attache les modules bâtis sur le champ ou préparés d'avance. (Fig. 3)

FIGURE 1

```
√ [F6] to include a field in the ANSWER; [F5] to give an Example   Main

PROFIL======No_Auteur===========No_Titre==========NomAuteur==========Profession=
        ||                  || BIBLIO          ||√                 ||
        ||                  ||                  ||                  ||
        ||                  ||                  ||                  ||
        ||                  ||                  ||                  ||

ARTICLE======No_Titre============Titre============Cat.d'art.=========1er lect.==
        || BIBLIO           ||√                 ||                  ||
        ||                  ||                  ||                  ||
        ||                  ||                  ||                  ||
        ||                  ||                  ||                  ||

ANSWER=========NomAuteur=============================Titre==================
    38 || Rousseau, Louis-Jean || Hanse, Joseph. Nouv. dictionnaire des difficul
    39 || Sauvageau, Philippe  || Développement de la bibliothèque de Québec
    40 || Savard, Réjean       || Etude du milieu et stratégie de promotion...
    41 || Tremblay, Clément    || L'évaluation des bibliothèques universitaires
    42 || Varloot, Denis       || Les évolutions technologiques récentes et...
```

FIGURE 2

```
                                                    Script: Fig2 --- Page 1

Esc (Scripts) (Play) (Figure1) Do_It! Menu (Image) (TableSize)
Up Up Up Up Up Up Up Up Up Up Up Up Up Up Up Up Enter End
UpImage UpImage Menu (Scripts) (End-Record)



                                                    Script: Figure1 --- Page 1

          Query

           Profil ¦ No_Titre ¦ NomAuteur ¦
                  ¦ _BIBLIO  ¦ Check     ¦
                  ¦          ¦           ¦
                  ¦          ¦           ¦

           Article ¦ No_Titre ¦ Titre  ¦
                   ¦ _BIBLIO  ¦ Check  ¦
                   ¦          ¦        ¦
                   ¦          ¦        ¦

          Endquery
```

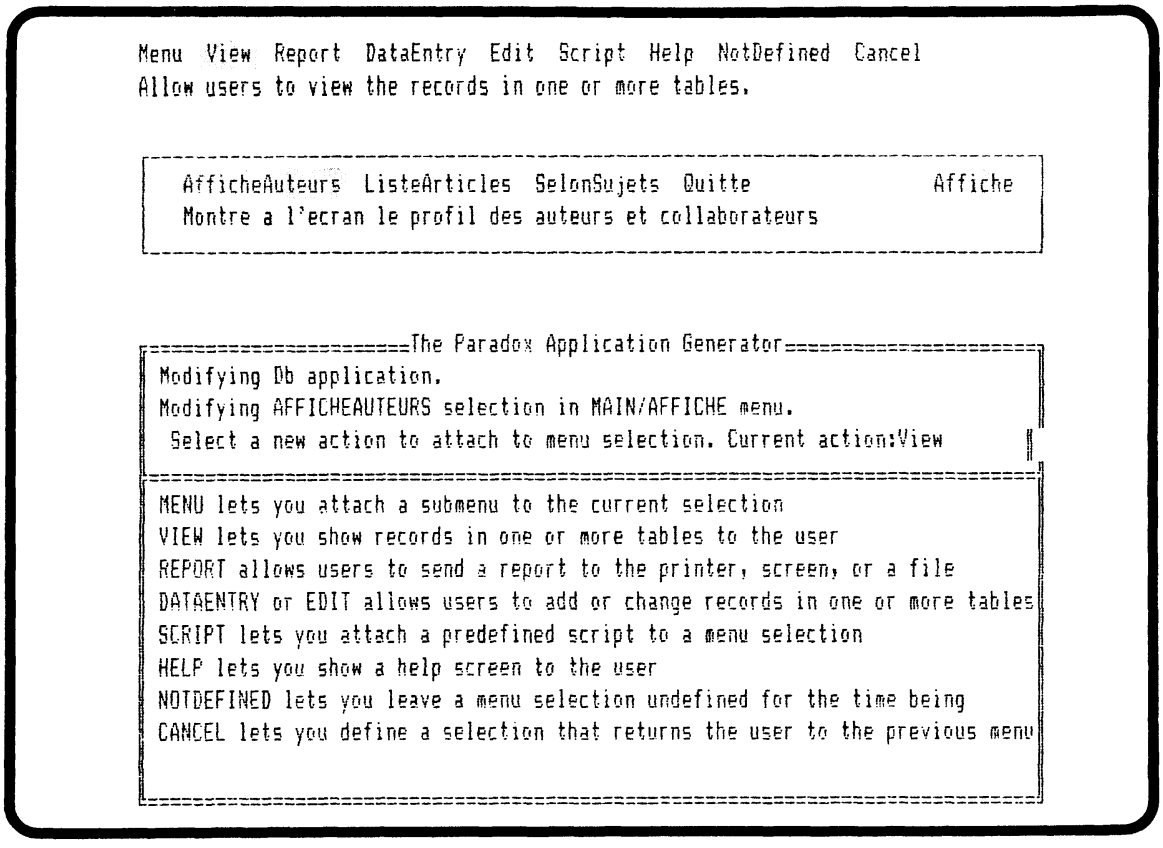
Paradox n'est pas la seule base de données relationnelle à offrir un générateur d'applications. R:base citée plus haut dans sa plus récente version, R:base System V, en propose un maintenant intégré à la base de données. DBASE III Plus s'est mis à la page en offrant l'aide de son Assistant tandis que deux compagnies indépendantes ont développé pour DBase des générateurs plus élaborés tels que Flash Code et Genifer. Ces automates de la programmation, d'abord réservés aux analystes pour réaliser rapidement un prototype, sont maintenant adaptés aux besoins du non-programmeur. Au dire de la revue *Business Software*, Paradox est «perhaps the best application generator yet developed for non programmers».[1]

Le générateur d'applications ne demande que quelques heures d'initiation. Il suppose toutefois une solide connaissance du système auquel il se rattache. Comme ApGen propose un menu qui sert à en produire un autre, celui de l'application, les mêmes termes sont utilisés tant lorsqu'ils s'adressent au créateur du logiciel qu'à son destinataire. Ainsi «Cancel» permet au concepteur un retour sur ses pas, ou accorde à l'utilisateur à travers le menu de l'application un rattachement à la fonction «Cancel», soit un retour au menu précédent. Cette ambiguité dans la terminologie n'est pas sans risque de confusion. Mais une fois les interactions entre les deux menus bien comprises, on s'accommode assez bien de ce vocabulaire passe-partout. Après deux ou trois séances faites à l'aide de l'exercice-guide fourni avec le programme Paradox, les obstacles sont levés et un premier logiciel d'auteur peut être apporté à l'atelier d'assemblage.

Les étapes sont alors les suivantes: rédaction des menus et des sous-menus destinés à l'utilisateur, arrimage à ces menus à chacun des fichiers ou groupes de fichiers, attribution de leur fonction telles que: affichage, impression, mise à jour, etc.; introduction des variables nécessaires à leur interrogation; présentation des écrans d'aide; et enfin mise en page de la bannière de présentation que les concepteurs ont surnommé «Splashscreen». L'opérateur intervient seulement

1 James E. Fawcette, «Speeding database design», *Business Software*, vol.4, no. 9 (September 1986), 10-17.

FIGURE 3



pour déclancher une opération, compléter un script, autoriser ou interdire un accès à l'utilisateur, et même passer outre à une mise en garde signalée par le système. «Do-it» et l'écriture automatique du programme est lancée. Elle défile sur l'écran. (Fig. 4)

Si après avoir fait jouer ce nouveau logiciel, des changements sont nécessaires, il suffit de remettre le générateur en marche et d'apporter les retouches là où elles s'imposent. Il est possible et facilement, de conserver les éléments valables d'un projet même dans la partie qui fait l'objet d'une révision.

Enfin, un autre jeu d'utilitaires permet de transcrire ce nouveau logiciel d'auteur dans une version exécutable «runtime» qui sera alors utilisable telle quelle par des tiers sur un micro de type compatible IBM, pourvu qu'il soit doté de 512K de mémoire et muni de deux unités de disquettes.

Un banc d'essai avec un premier logiciel d'auteur: DB

Pour illuster mon propos, j'ai créé, sans programmation, le prototype d'une base de données pour la gestion de *Documentation et bibliothèques*. Ce logiciel aura comme utilité d'assister le Comité de rédaction dans la tenue à jour des sommaires de chaque numéro et dans l'établissement du calendrier de publication. De plus, il sera un instrument d'analyse rétrospective de la revue. On interrogera cette base sur les thèmes abordés et le profil de ses auteurs et collaborateurs. L'analyse permettra au Comité de rédaction de veiller à l'équilibre du contenu de la revue. Sa grille d'analyse est empruntée de l'étude de Lajeunesse et Wilson sur l'analyse des vingt-cinq ans de *Documentation et bibliothèques* pour l'établissement des catégories d'articles et de leur provenance[2]. Pour les fins de cette illustration, je retiendrai seulement les rubriques les plus caractéristiques de la base de données.

2. Marcel Lajeunesse et Lise Wilson, «Vingt-cinq ans de publication périodique en bibliothéconomie au Québec: analyse quantitative du Bulletin de l'ACBLF/Documentation et bibliothèques (1955-1979)», *Documentation et Bibliothèques*, vol.27, no 2 (Juin 1981), 53-67

FIGURE 4

```
; Db2.sc

x2 = "AfficheAuteurs"
while (true)
 escape = false

 Clear
 ShowMenu
    "AfficheAuteurs": "Montre a l'ecran le profil des auteurs et collaborateurs",
    "ListeArticles": "Montre a l'ecran l'etat des articles publies et a paraitre",
    "SelonSujets": "Signale les articles et recensions par themes",
    "Quitte": "Retourne au menu principal"
  Default x2
  To x2

 switch
  case x2 = "AfficheAuteurs" :
   if (istable("Answer")) then
    Delete "Answer"
   endif
   Play "Dbq5"  Do_It!
```

Cette base comportera trois fichiers, un premier sur les auteurs (PROFIL), un deuxième sur les articles (ARTICLE) et un troisième sur les thèmes traités (SUJET). (Fig. 5)

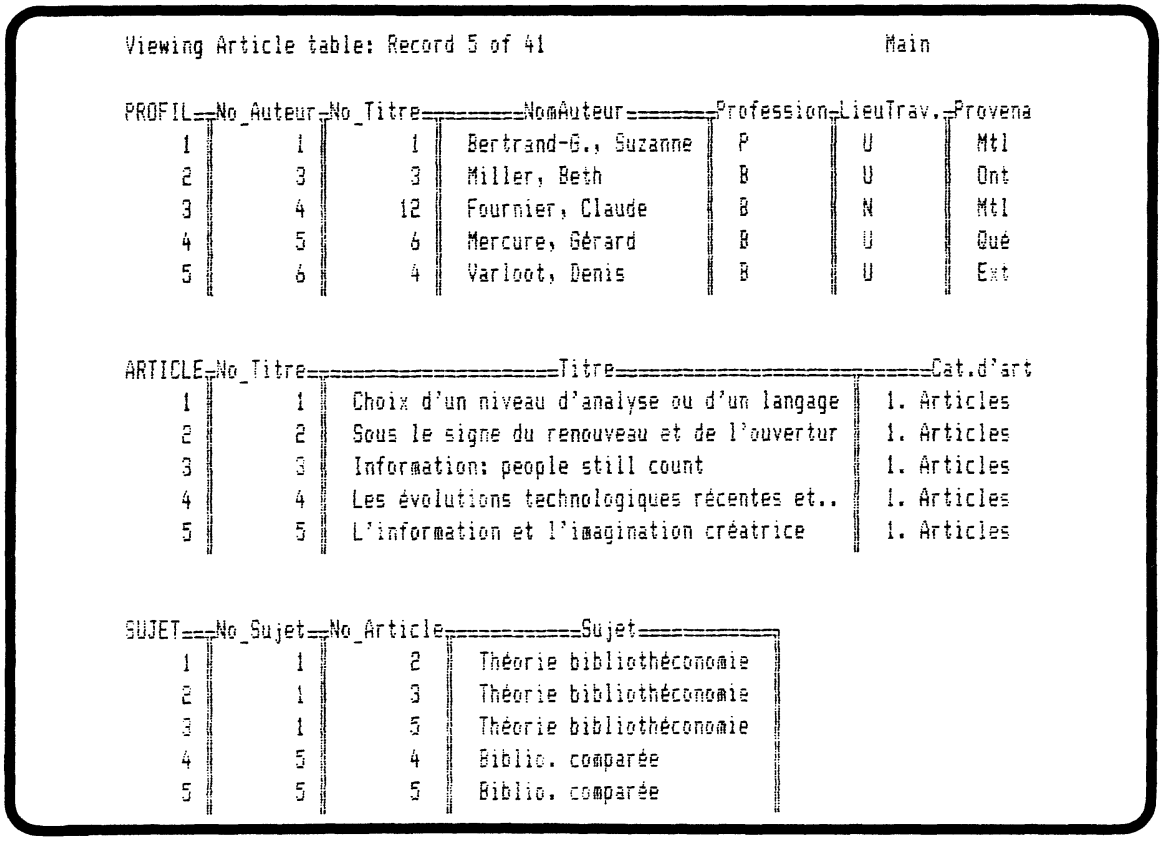
En établissant les relations appropriées entre ces trois fichiers par des index et des scripts, il sera possible après l'accumulaton des données pendant quelques années d'établir la provenance des auteurs selon leur milieu de travail, leur appartenance à un groupe professionnel et leur distribution géographique. Pour l'année 1986, cette base DB donne déjà un apercu de la production littéraire et le palmarès des sujets en vogue (à l'exclusion des recensions d'ouvrages et de la catégorie «Autres» qui comprend l'index annuel). (Fig. 6)

Ce logiciel servira dans la gestion courante à tenir un calendrier de parution des comptes rendus par rapport à la date d'édition de l'ouvrage, à maintenir à jour une banque de collaborateurs éventuels et leurs coordonnées, etc. L'utilisation première sera toutefois la mise à jour automatique des sommaires des numéros à paraître selon la disponibilité des articles et l'équilibre des numéros. Il faut dire ici qu'une livraison de *Documentation et bibliothèques* prend forme progressivement un an à l'avance. En dépit de cette planification, il est difficile de respecter les échéances. À titre d'essai, voyons ce que donne un tel logiciel fait sur mesure pour la production automatique d'un tel rapport. À remarquer qu'un seul article n'est pas encore rentré malgré la date de tombée déjà tardive... (Fig. 7)

Possibilités et limites

Comme le démontre cet exemple, le produit peut être honnête même venant d'un informaticien naïf. Le générateur d'application impose quelques contraintes à sa spontanéité dans l'écriture des scripts et dans l'utilisation notamment de rubriques calculées. Le non-programmeur apprend vite cependant à ruser avec le système en complétant le script en dehors du générateur pour ajouter ces calculs et compléter des scénarios. Le système est assez tolérant. Il préviendra à l'occasion dans l'enchaînement de fichiers, d'un risque d'incohérance et acceptera de passer outre à la mise en garde. Il permet les retouches sans devoir tout reprendre. Mais attention! Il peut tout laisser en plan si à la suite d'un changement dans la structure d'un fichier un script n'a pas été mis à jour avant ou pendant la séance de développement sous

FIGURE 5



Viewing Article table: Record 5 of 41 Main

PROFIL

No_Auteur	No_Titre	NomAuteur	Profession	LieuTrav.	Provena
1	1	Bertrand-G., Suzanne	P	U	Mtl
2	3	Miller, Beth	B	U	Ont
3	4	Fournier, Claude	B	N	Mtl
4	5	Mercure, Gérard	B	U	Qué
5	6	Varloot, Denis	B	U	Ext

ARTICLE

No_Titre	Titre	Cat.d'art
1	Choix d'un niveau d'analyse ou d'un langage	1. Articles
2	Sous le signe du renouveau et de l'ouvertur	1. Articles
3	Information: people still count	1. Articles
4	Les évolutions technologiques récentes et..	1. Articles
5	L'information et l'imagination créatrice	1. Articles

SUJET

No_Sujet	No_Article	Sujet
1	2	Théorie bibliothéconomie
2	3	Théorie bibliothéconomie
3	5	Théorie bibliothéconomie
4	4	Biblio. comparée
5	5	Biblio. comparée

ApGen. Il faut laisser la partie non revisée ou «undefined» avant de lancer la programmation automatique. Mais c'est la seule occasion où il oblige le profane à une prévoyance systémique de programmeur. Et je le dis d'expérience...

Au cours de cet essai, j'ai dû de plus faire face au piège classique qui guette tout apprenti. L'espace disponible sur le disque est vite devenu encombré par mes esquisses laissées en plan et mes fichiers dispersés aux quatre coins du disque (hum!) ce qui ralentissait l'exécution des scripts. Un logiciel monté avec ApGen prend de l'espace. À partir des trois fichiers de départ, il en a construit douze autres. Les sept scripts au menu ont été accompagnés d'une vingtaine d'autres générés par le système. Il m'a donc fallu d'abord faire du rangement, remettre sur le chantier une nouvelle version, la première étant, comme le veut la pratique, bonne à mettre au panier. Une approche automatique ne dispense pas d'une organisation méthodique du travail.

Avec Paradox, il n'est pas possible, sans programmation, d'obtenir une notice bibliographique à deux auteurs sans devoir répéter la notice pour chacun d'eux. La répétition d'une rubrique peut être évitée seulement dans la production d'un rapport lorsqu'elle détermine le classement d'un sous-groupe de documents. Cette forme de présentation convient bien à l'information de nature commerciale, l'exemple classique étant la gestion des fichiers de clients, de produits et de commandes. Et c'est dans ces mêmes types d'applications qu'il faut utiliser à profit ce genre de base de données dans les bibliothèques. Espérer davantage serait vouloir faire d'une base de données relationnelle une base de données bibliographique. La disposition bibliographique a d'autres exigences que seule une programmation plus élaborée peut satisfaire. Le langage spécialisé de programmation, Paradox Application Language (PAL) permet de telles modifications.

L'informaticien rompu à la programmation trouve qu'ApGen impose des limites trop sévères. L'exécution du programme est relativement lente. En cours de révision, celui-ci rejette les instructions ajoutées au clavier à moins d'avoir pris la précaution de renommer ou de sauvegarder à part; l'écriture d'un scénario dans cette forme standardisée comporte des restrictions: interdiction entre autres d'utiliser les opérateurs logiques (et) et (ou), d'introduire des calculs. Pour les analystes Alperson et King de Data Base Advisor, ApGen est «acceptable for only small applica-

tions»[3]. C'est un fait dans la mesure où l'on s'en tient au cadre imposé par le générateur sans profiter de la possibilité offerte au menu d'improviser ses propres scripts ou scénarios. Cela demande alors une plus grande maîtrise de Paradox et suppose un recours éventuel au langage PAL, mais au profit d'une plus grande liberté de mouvement.

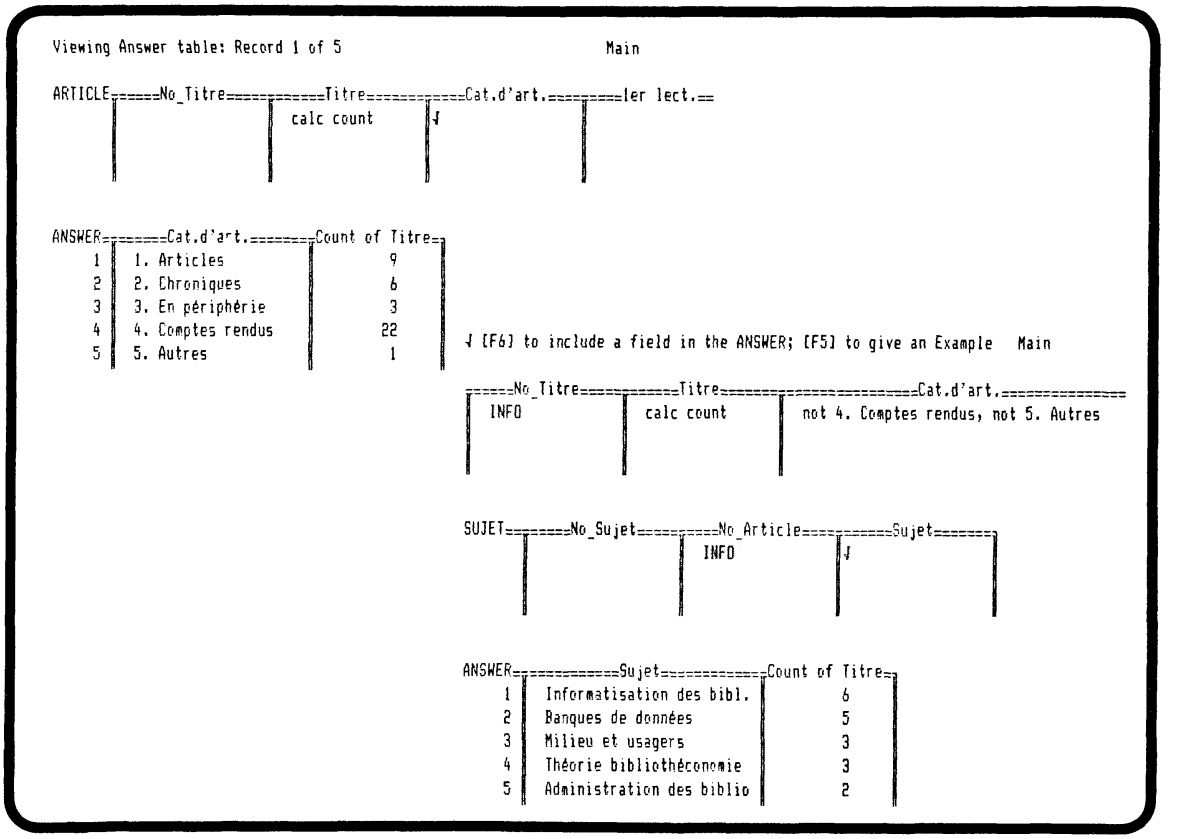
Ce qu'on gagne en simplicité en confectionnant un logiciel d'auteur, on le perd en souplesse. L'utilisateur est cantonné aux seuls scénarios d'interrogation de la base tels qu'établis sous ApGen. Par contre l'initiation au logiciel d'auteur est beaucoup moins exigeante: il suffit de lire la commande proposée au menu et de s'y conformer. Point besoin de s'initier à Paradox. On peut ainsi proposer au personnel d'un service d'une bibliothèque d'alimenter une base particulière sans devoir investir dans l'apprentissage du système ou d'installer Paradox sur plus d'un micro-ordinateur.

Paradox et ApGen sont d'origine américaine. Pour la création d'un logiciel en français, ils affichent les messages à l'intention de l'utilisateur en anglais lorsqu'ils sont produits automatiquement. On peut donc avoir, juxtaposé à «Affiche» et «Imprime», un «Leave» qui détonne. Il est possible de corriger cette anomalie par une retouche au programme en substituant «Leave» par «Quitte» ou «Fin». Plus grave, ApGen n'accepte pas toujours les accents! C'est le cas des menus et des scripts. Je n'ai pas trouvé de façon simple de corriger cette lacune. Ansa Software, producteur de Paradox, annonce une version française de Paradox sous peu, ce qui éliminera d'emblée ce problème d'affichage bilingue...

La programmation disparaît mais l'analyse demeure

Cette chronique pourrait s'arrêter ici avec cette rapide démonstration d'un logiciel fascinant. Mais le problème de fond n'est pas réglé pour autant.

FIGURE 6



3. Jay Alperson and Steve King, «Paradox 1.1: More power and less pain», *Databased Advisor*, vol.4, no. 12 (December 1986), 12-21.

Le travail d'analyse ne se fait pas automatiquement. Or une base de données relationnelle exige une grande rigueur dans l'organisation et la gestion des fichiers, afin de contrôler la redondance, d'éviter l'incohérence et d'optimiser le rendement. Il existe dans de nombreux ouvrages des méthodes et organigrammes pour établir les bonnes relations et les valider. Il n'est pas nécessaire de connaître l'algèbre relationnelle d'un système d'information pour pouvoir bâtir convenablement une base de données relationnelle. Pour des projets peu élaborés, il suffit d'appliquer un certain nombre de règles simples de normalisation des relations que l'on retrouve dans les ouvrages de micro-informatique. Un ouvrage traitant de Paradox, qui à pour titre *Paradox for the Programmer,* comporte un chapitre sur l'analyse et la conception de système[4]. Si le projet prend de l'envergure, il vaut mieux le confier à un informaticien de carrière.

FIGURE 7

2/03/87 D O C U M E N T A T I O N E T B I B L I O T H E Q U E S Page 1

Sommaire du numéro du 1-Oct-86

Date de tombée: 1-Nov-86

TITRE OU THEME DE L'ARTICLE	AUTEUR DEMANDE	DEMANDEUR	2ième LECT.	ACCEPTE	RECU
1. Articles					
L'informatisation des b. : progrès ou menace	David-McNeil,Jeanine	G.D.	G.B.	o	o
Télédéchargement et création de bases de données..	Henry-Rousseau, Y.	G.D.	C.F.	o	o
2. Chroniques					
La logithèque de la bibliothèque de Québec	Payeur, Jean	G.B.	J.D.	o	o
Création d'une banque de données: oeuvres d'art...	Gauthier, Michel	N.L.	G.M.	o	o
3. En périphérie					
La micro-informatique automatique	Mercure, Gérard	N.L.	A.L.	o	n
4. Comptes rendus					
(Documentor) Lexique d'information documentaire	Bernier, Gaston	C.F.	N.L.	o	o
(U.Eco) De bibliotheca	Brault, Jean-Rémi	J.D.	C.F.	o	o
Charbonneau, H. Livres en langue française pour...	Bergeron, Louisette	G.B.	N.L.	o	o
Les guides de films de la ville de Mtl et Cégep...	De Chevigny, Louise	J.D.	G.B.	o	o
5. Autres					
Index annuel 1986	Malette, Danielle	N.L.	A.L.	o	o

N. B. La date de tombée est la date à laquelle les textes CORRIGES doivent parvenir à la secrétaire à la rédaction. Demander aux auteurs de faire parvenir leurs textes au moins un mois avant cette date.

4. Nelson T. Dinerstein, «Advanced design principles», in *Paradox for the programmer.* Glenview, Ill., Scott, Foresman, 1986, p. 189-217.

Car il faut, avec ces générateurs, se garder de lancer un projet trop ambitieux ou qui sort du gabarit proposé, tel que l'affichage, la mise à jour de fichiers et l'impression de listes. Une fois plus habitué, il est possible de ruser avec le système et de modifier légèrement le programme en procédant par emprunts et imitation. À la condition alors d'y mettre du temps, car il faudra alors acquérir par essais et erreurs ses premières notions de programmation.

S'initier à Paradox et à son générateur ne demande que quelques heures. En exploiter tous les raffinements peut réclamer plusieurs mois de pratique. S'il est un premier paradoxe, celui d'offrir un progiciel à la fois puissant et facile d'utilisation, il en est un second qu'il faut bien admettre: le mode d'emploi est bien imposant — près de 400 pages — pour un logiciel dit facile. Le mérite de Paradox réside plutôt dans le caractère progressif de l'initiation à ses commandes et à son langage de programmation. Le niveau d'apprentissage peut être dosé en fonction de ses besoins de gestion. Le langage PAL permettra d'aller au-delà d'une application immédiate en faisant appel cette fois aux ressources de la programmation.

Ces nouvelles bases de données n'exigent pas de programmation mais réclament des instructions de la part de l'utilisateur pour accomplir une tâche définie. Les menus à choix multiples et les macro-commandes permettent d'y arriver. Au lieu d'appliquer les règles d'un langage de programmation, l'utilisateur remplit en quelque sorte un questionnaire; le menu est le guide commenté de cette batterie de questions. Pour réduire le nombre des prescriptions occasionnées par cette façon de faire, il faudrait plus d'intelligence encore. La faible puissance des ordinateurs personnels contraint pour le moment les développeurs de systèmes à recourir à des règles simples d'IA. Déjà Paradox et ApGen réunis, réclament toute la puissance d'un PC. Il leur faut obligatoirement une mémoire de 640K, et pratiquement la présence d'un disque dur.

Avec l'apparition de ces bases douées d'un peu d'intelligence et de ces générateurs automatiques de programmes, les bibliothécaires peuvent néanmoins utiliser dès maintenant ces progiciels dans la gestion quotidienne des données de leur service. À la condition de consentir le temps nécessaire à l'apprentissage du système et au développement de leurs logiciels d'auteur. Manipulation de statistiques, administration de dossiers, tenue de répertoires d'adresses, production de listes de toutes sortes, tableaux de bords et de répartition des tâches d'un projet, telles sont les applications premières de ces programmes automatiquement ajustables aux besoins de chacun.

La réalisation d'un logiciel d'auteur sans programmation ne dispense pas du travail de création et c'est aussi ce qui demande du temps. Que l'on songe, en audiovisuel, par exemple, aux heures de préparation nécessaires à la mise en oeuvre du plus modeste des diaporamas... Et comme pour ces diaporamas, les logiciels d'auteurs les plus convaincants ne sont pas ceux dont le montage est le plus savant, mais ceux qui répondent brillamment à la question de l'heure.

Gérard Mercure
Directeur de la bibliothèque
Université du Québec à Rimouski