



**HAL**  
open science

# Enseignement des logiciels et de la programmation avec logiciels

Caroline de Schaetzen

► **To cite this version:**

Caroline de Schaetzen. Enseignement des logiciels et de la programmation avec logiciels. Troisième rencontre francophone de didactique de l'informatique, Jul 1992, Sion, Suisse. pp.129-137. edutice-00359225

**HAL Id: edutice-00359225**

**<https://edutice.archives-ouvertes.fr/edutice-00359225>**

Submitted on 6 Feb 2009

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

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

# ENSEIGNEMENT DES LOGICIELS ET DE LA PROGRAMMATION AVEC LOGICIELS

Caroline de SCHAETZEN

## 1. INFORMATIQUE ET ENSEIGNEMENT DE LA TRADUCTION

L'arrière-plan de notre expérience de sept années en didactique de logiciels à 100 étudiants par an en moyenne est celui des outils informatisés d'aide à la traduction. Une brève évocation de ce décor, basée sur plusieurs questionnaires<sup>1</sup>, limitera sans doute les généralisations qui pourraient être faites du présent bilan.

Loin de l'image d'Epinal de la traduction entièrement automatique, les traducteurs disposent d'une plume électronique, le traitement de texte, auquel s'adjoignent : un télécopieur, un gestionnaire de glossaires<sup>2</sup> (ou gestionnaire de fichiers permettant la consultation de dictionnaires sans sortir du traitement de texte, avec notamment la possibilité d'insérer, par la pression d'une touche, de l'équivalent en langue étrangère d'un terme au sein du texte en cours de traduction), des banques de terminologie (ou dictionnaires spécialisés) accessibles par les serveurs documentaires ou sur CD-ROM.

Les professionnels de la « transposition » sont désireux de se recycler en informatique de la traduction. Les expériences malheureuses ont en effet été peu nombreuses<sup>3</sup> parmi eux. Les plus exigeants d'entre eux souhaiteraient que les jeunes traducteurs connaissent deux traitements de texte, le DOS (y compris les procédures de récupération de fichiers) et le fonctionnement de l'ordinateur (choix et installation de cartes d'extension, par exemple). La connaissance de la *post-édition*, à savoir la correction... des mauvaises traductions automatiques, n'est par contre réclamée que par les vendeurs de systèmes de traduction automatique

---

<sup>1</sup>. un questionnaire distribué pendant trois ans au début et à la fin du cours d'informatique de première licence en traduction, à l'Institut Libre Marie Haps; une réunion des six professeurs d'informatique d'un PIC (réseau d'échange d'étudiants) ERASMUS (programme pour la mobilité étudiante de la CEE); un questionnaire aux maîtres de stage des étudiants en traduction sur leur souhait en matière de formation informatique des étudiants; une enquête auprès d'un échantillon représentatif des traducteurs belges, portant sur le matériel et les logiciels utilisés par eux et sur leur connaissance de la traduction automatique; des entretiens approfondis avec des interprètes

<sup>2</sup>. Quelques interprètes utilisent également les gestionnaires de glossaires, sur un ordinateur portable, pour chercher l'équivalent d'un terme mais uniquement si l'interprétation en cabine se fait à deux, sinon l'interprète n'a le temps de consulter ses listes que pendant les pauses entre les conférences, donc à des moments où cette consultation lui est pratiquement inutile

<sup>3</sup>. abandon de 3 000 fiches de vocabulaire avec définition et commentaires, faites sur traitement de texte, en raison de la lenteur croissante de la consultation du fichier, et retour aux fiches en carton "boîte à chaussures", par exemple

surannés, dont ce « chantage à la modernité » vise à faire acquérir par les établissements d'enseignement le système de traduction en question...

Traitement de texte, gestionnaires de glossaires et DOS sont enseignés dans toutes les facultés et instituts européens d'enseignement de la traduction, à raison de 45 à 150 heures/semestre et d'un micro-ordinateur pour 10 étudiants environ. Certains établissements y joignent l'interrogation des banques de terminologie, celle des banques documentaires, de la messagerie électronique et, plus rarement, de la programmation. La formation des professeurs est pour moitié celle de l'informatique « classique » (de gestion) et pour moitié une auto-formation de professeurs de langues ou de traduction ; c'est qu'il n'existe qu'un établissement européen<sup>4</sup>, à dispenser une formation (maîtrise) à l'informatique linguistique et depuis deux ans, à la traduction automatique. Le recyclage permanent de ces professeurs d'informatique est ardu : outre la nécessité d'étudier une nouvelle version de logiciel tous les trois ans au moins, ils se voient à présent astreints, pour ceux qui ont enseigné la programmation impérative, de se former aux disciplines qui constituent l'informatique linguistique (derniers développements de la linguistique, logique, intelligence artificielle, éléments des sciences cognitives, etc.) et à la programmation récursive. Ils doivent également fonctionner comme info-centres, notamment lors de l'acquisition par les étudiants de leur propre ordinateur.

Les étudiants belges s'informent en effet de plus en plus et de plus en plus tôt (au début de la 3<sup>e</sup> année d'étude et non plus à la fin de la 4<sup>e</sup> et dernière année) ; le micro-ordinateur qu'ils choisissent est le plus souvent un produit blanc, avec un traitement de texte piraté. Le cours explique l'informatisation de l'étudiant : tant le nombre d'heures de cours sur machine dispensés au graduat en psychologie, logopédie, audiologie de l'Institut Marie Haps que le nombre de micros achetés par les étudiants suivant ces graduats est moindre, alors que l'informatisation de la profession est à un stade voisin de celui de la traduction...

L'intégration des outils dans la didactique même des langues étrangères et de la traduction/interprétation est faible : d'une part, les outils pédagogiques sont toujours moins utilisés dans l'enseignement supérieur que dans le secondaire et de l'autre, les professeurs (féminins, pour une bonne part) sont déjà aux prises avec d'autres arsenaux techniques en évolution rapide : caméras vidéo, laboratoires de langues, salles d'interprétation ; en outre, il n'existe pratiquement pas de didacticiel pour l'enseignement avancé des langues ; enfin, en raison de la rareté du matériel, priorité a été donnée partout à la préparation de l'étudiant aux outils informatisés de sa future profession. Aussi, les seules activités organisées sur machine sont souvent la préparation des traductions et du mémoire par les étudiants, l'élaboration et la consultation de glossaires, ainsi que l'organisation récente de certains cours de traduction dans la salle d'informatique (avec le problème de maîtrise des multiples logiciels requise par le professeur de traduction pour donner ce cours sans le professeur d'informatique). Des collaborations s'amorcent : dans le cadre des échanges d'étudiants ERASMUS, un professeur belge, français, hollandais,

---

<sup>4</sup>. l'UMIST à l'Université de Manchester

allemand et espagnol se concertent actuellement pour le choix d'un format de fichier d'échange de vocabulaire et de textes (à traduire).

## **2. ENSEIGNER LES LOGICIELS**

Les logiciels méritent à juste titre l'intérêt des pédagogues et cet intérêt sera, il faut l'espérer, à l'origine de rétroactions fructueuses : pour le traducteur, la résolution des problèmes d'affichage des caractères russes en traitement est par exemple plus productive qu'une avancée en traduction automatique, forcément chère et timide<sup>5</sup>. Autre exemple : le Ministère responsable des technologies nouvelles a commandé une recherche à l'Institut Marie Haps sur l'utilisation des index d'ouvrages techniques rédigés en traitement de texte, afin d'accélérer l'élaboration des dictionnaires spécialisés : l'écriture d'un programme qui consulte un index et recopie les définitions ou une phrase-exemple comportant le terme neuf est facile et beaucoup moins coûteuse que les systèmes intelligents de dépouillement automatique de documents, qui requièrent un module d'analyse syntaxico-sémantique complet des phrases, un programme de statistique complexe et un système expert (dont le corpus de règles varie d'un domaine à l'autre). Ces exemples justifient l'enseignement des logiciels et la cotation de cette connaissance dans l'enseignement supérieur préparant directement à une profession.

### **2.1. Simplification des logiciels**

Les logiciels tendent à se simplifier. WINDOWS est la plate-forme d'écriture des logiciels et l'interface conviviale que le monde DOS envoyait à l'univers Macintosh. De plus, les logiciels dédiés laissent progressivement la place à des sous-logiciels spécialisés : les logiciels de traitement de texte multilingues disparaissent devant les nouveaux alphabets cyrilliques et hébraïques de WORDPERFECT ou de MS WORD<sup>6</sup> : ils ne pouvaient égaler la puissance et l'étendue des commandes de traitement de texte de ces grands standards.

### **2.2. Enrichissement des logiciels**

C'est que les grands logiciels croissent rapidement : chaque version du manuel de traitement de texte pour MS WORD double de volume par rapport au précédent<sup>7</sup>. Cet enrichissement semble s'accélérer : l'intervalle entre les ajouts mineurs à une version de logiciel (entre une version appelée 1.1 et 1.2, par exemple) était d'un an. A présent, ce que sépare cet intervalle, ce sont des ajouts conséquents et des remaniements (c'est le premier chiffre des versions qui change et ce n'est pas un artifice commercial).

D'où la nécessité d'enseigner de nouveaux savoir-faire : comme la publication assistée est possible avec et facilitée par le traitement de texte, les

---

<sup>5</sup>. la TA dépend des développements de la linguistique, de l'informatique et, comme elle traite de sujets spécialisés, de ceux de l'intelligence artificielle (systèmes experts)

<sup>6</sup>. CYRIL, d'Apsylog

<sup>7</sup>. Guide complet de WORD 4, Guide complet de WORD 5, Guide complet de WORD 2 pour WINDOWS. C. de Schaetzen, Eyrolles, Paris.

bureaux de traduction se sont rapidement transformés en agences de traitement de toute la chaîne de l'information documentaire, depuis la conception du texte (adaptation dans une autre langue d'une bande sonore en sous-titres ou d'un texte publicitaire en publicité en passant par la publication de dépliants traduits). Certaines « branches mortes <sup>8</sup> » ne devraient par contre plus être enseignées, sauf à titre d'intérêt historique : les gestionnaires de glossaires ont fait disparaître une technique de compilation de dictionnaires basée sur des manipulations de fiches de carton et les vérificateurs orthographiques devraient relativiser la cotation des fautes d'orthographe, y compris dans les langues étrangères...

La rapidité des mises à jour est parfois source d'incohérences d'une couche de logiciels à l'autre : les versions successives du sommet de la pyramide (traitements de texte, gestionnaires de bases de données, logiciels intégrés) sortent un an après celles du niveau intermédiaire (WINDOWS), lesquelles doivent attendre le bas de l'échelle (DOS). Sans parler des logiciels interfacés avec une pièce de ce savant échafaudage (la version « réseau » d'un gestionnaire de glossaires, qui doit être compatible avec les grands traitements de texte du marché, qui vient d'être commercialisée doit déjà être réécrite pour WINDOWS) et des produits dérivés (conversions successives des dictionnaires tournant sur ledit gestionnaire de glossaires, incompatibilité de programmes résidents, etc.).

Les logiciels sont sous-utilisés parce que sous-enseignés : leur étendue <sup>9</sup> en compromet la mémorisation. Non pas tant celle de la syntaxe des commandes (le travail, y compris l'examen, avec un manuel doit être permis dans l'enseignement comme il l'est dans la vie professionnelle) mais de l'existence même d'une commande automatisant telle ou telle opération. Un exposé détaillé du parti que peut tirer le futur professionnel des commandes doit être fait par le professeur : l'organisation du travail « autour » d'un logiciel central fait partie de l'enseignement des logiciels. L'enseignement systématique d'un logiciel est donc peut-être préférable, malgré la difficulté qu'éprouvera le jeune diplômé s'il doit utiliser dans sa vie professionnelle, parfois simultanément, un ou plusieurs autres marques de logiciel.

Le rythme d'apprentissage peut également faciliter la mémorisation. Notre observation d'adultes en recyclage au traitement de texte nous a fait préférer le découpage de la matière en trois ou quatre phases, séparées par six mois d'utilisation en situation réelle. C'est également la raison pour laquelle les recyclages étalés en cours hebdomadaires favorisent plus la rétention à long terme que les sessions intensives (même limitées à des sessions de 4 heures).

Un tri s'impose en outre pour ce qu'il convient de montrer, d'exposer, d'enseigner, d'exercer et de coter : apprendre aux secrétaires en recyclage un gestionnaire de bases de données, un intégré et un traitement de texte serait

---

<sup>8</sup>. l'expression est de L. de Brabandère, exposé au colloque sur la promotion de l'enseignement technique organisé par les associations de parents de l'enseignement secondaire à Bruxelles en juin 1992 (*non publié*)

<sup>9</sup>. le glossaire des touches, des commandes et du vocabulaire du traitement de texte de la version 2 de WORD pour WINDOWS est d'environ 2.500 entrées

irréaliste et se ferait au détriment de la maîtrise de ces outils : un DBASE IV doit-il être inculqué à une secrétaire qui ne procède qu'une fois par an à un tri multicritère ? Ne peut-on lui « bricoler » un programme en traitement de texte ? Autre exemple : le maniement des mailings doit-il être enseigné à une dactylo : il suppose assimilées les notions de variable et les conditions (avec alternative), qui relèvent de la programmation.

### 2.3. Coût des logiciels

Si on le compare à celui du matériel, des compilateurs et des interpréteurs, le prix des logiciels ne baisse pratiquement pas. L'inexistence de conditions d'achat groupé, qui a longtemps prévalu pour les établissements d'enseignement et le sont encore pour les logiciels à diffusion relativement restreinte, font des mises à jour de logiciels un poste élevé dans le budget des établissements d'enseignement, quand elles ne les pousse pas elles-mêmes au piratage...

Ce coût doit être mis en rapport, devant les étudiants, avec les apports réels de certains logiciels et de leurs produits : les dictionnaires multilingues commercialisés avec les gestionnaires de gestionnaires mais surtout certaines banques de terminologie accusent un décalage rédhibitoire sur l'usage, par rapport à ceux qui sont publiés sur papier : c'est que ce sont ces dictionnaires imprimés qui constituent le *corpus* de ces banques!

## 3. ENSEIGNER LA PROGRAMMATION PAR LANGAGE-MACRO

### 3.1. Vertus pédagogiques des langages-macros

La programmation est un savoir-faire et un jeu qui relèvent non seulement de la culture générale mais aussi de la culture et même de la compétence du traducteur : « programmer, c'est passer d'un problème sur des objets du monde au texte d'un programme exécutable par un langage informatique donné » (C. Pair).

Ce sont surtout des impératifs de rentabilité immédiate qui militent en faveur de la didactique de la programmation. Pour les traducteurs indépendants, le coût des programmes faits sur mesure est prohibitif, même pour leurs applications administratives (facturation, gestion de traductions). Le traitement de texte est la plume des métiers de l'écriture et aucun programme « autour » du traitement de texte n'existe à notre connaissance sur le marché, pour quelque profession que ce soit (on vend uniquement quelques feuilles de style). Or de simples macro-commandes sont déjà utiles aux traducteurs : l'exécution des commandes (compilées) qu'elles comportent est puissamment accélérée, si on les compare au déroulement des programmes interprétés d'un DBASE III Plus. Les rédacteurs ont souvent besoin de programmes ponctuels (mise en page d'un ouvrage prêt à cliquer selon une maquette d'édition, ajout d'une insistance à une phrase par l'adjonction de la formule : *c'est...que*), qui doivent donc pouvoir être écrits rapidement.

La programmation avec macro-commandes se fait pour une part par enregistrement, ce qui facilite grandement l'écriture des instructions : il suffit d'ajouter quelques boucles conditionnelles au texte du programme ainsi écrit automatiquement pour obtenir un programme performant. La différence entre

commandes à exécution différée (les instructions) et interactive peut également être montrée pendant l'apprentissage du traitement de texte lui-même. La transition entre le traitement de texte, qui prépare à la programmation, et la programmation d'applications complètes destinées à des tiers est donc progressive pour les futurs traducteurs ou interprètes et à tous les stades de leur apprentissage de la programmation, les programmes qu'ils produisent sont directement utilisables.

Les langages de macro-commandes du traitement de texte, par exemple, assistent la programmation : l'exécution des programmes peut se faire au ralenti (plusieurs degrés de ralentissement), ce qui, avec les messages d'aide, facilite le déblocage. Même chose pour l'affichage en écho du résultat des instructions du programme, qui peut être activé pendant la phase de test. Enfin, les commandes fréquentes et les noms de touches à transcrire peuvent être eux-mêmes stockés sous forme de macro-commandes. Il n'est bien sûr pas obligatoire de ne mettre qu'une instruction par ligne et de numéroter ces lignes. De plus, l'affichage d'une macro (même au beau milieu d'un document) et sa correction en mode plein écran sont facilitées par les fonctions de traitement de texte, bien entendu disponibles en tout temps (recopies rapides de fractions de macro par utilisation des fonctions « glossaire » et de la mémoire-tampon de texte, recherche et remplacement global de portions de texte de programmes) ; le stockage des macros est également confortable (elles sont dans le même fichier que les bibliothèques de paragraphes standards, donc accessibles en tout temps ou, si l'utilisateur préfère, elles sont associées à certains types de texte. De plus, les fenêtres permettent l'affichage du texte d'une macro en même temps que son test dans une seconde fenêtre. Les tests d'une portion de programme sont aisés (il suffit de sélectionner la partie à tester dans le texte du programme, de la recopier en lui donnant un autre nom que le programme lui-même, de lancer cette nouvelle macro et éventuellement de la corriger avant de la replacer dans le programme complet).

Le langage des macro-commandes est assez complet (une macro peut appeler des routines ou être récursive, par exemple) et surtout, son niveau est très élevé, puisqu'il permet l'inclusion de commandes du logiciel lui-même : finis les programmes peu motivants des débutants en BASIC, qui impriment laborieusement le résultat d'une addition.

Le langage des macro-commandes est convivial : sa syntaxe (française, où le nom des touches est en toutes lettres) les rapproche des pseudo-langages. Les langages de macros ne provoquent pas de réactions de défense contre la programmation chez les débutants peu portés vers les démarches logiques, en raison de la convivialité des logiciels avec lesquels ils sont couplés et de l'intérêt que ces logiciels ont déjà éveillé chez l'apprenant. C'est important pour les esprits « littéraires » peuplant les établissements d'enseignement des métiers de diffusion (traduction, documentation, journalisme, secrétariat). Certaines macros peuvent en outre s'exécuter dès le chargement du traitement de texte lui-même. L'apprentissage peut être progressif : en créant une fenêtre temporaire dans son programme, le débutant peut y effectuer des traitements compliqués sur des entités logiques, par exemple une modification de fiche de vocabulaire (changement de place de rubriques, etc.) ; après cette « cuisine », la fiche modifiée peut être recopiée dans le

fichier d'origine, dans la fenêtre principale. Cet artifice évite le recours aux variables indicées et à l'imbrication multiple de boucles complexes.

Les langages de programmation par macro-commandes sont fonctionnels, donc proches des traitements logiques humains. Certains établissements de formation à la traduction enseignent la programmation récursive d'applications de traduction automatique mais ils ne peuvent dépasser les rudiments car cette démarche est complexe. La programmation d'applications de linguistique informatique (en PROLOGUE, le plus souvent) n'est enseignée que dans des facultés de linguistique et d'ingénierie<sup>10</sup> ; c'est en effet là que sont conçus les programmes de traduction automatique, complexes et toujours au stade de la recherche. Pour les traducteurs humains, l'effort requis par un tel apprentissage est disproportionné par rapport aux résultats : l'utilisation d'un programme de traduction automatique ne requiert pas la connaissance de son fonctionnement ; enseigner la programmation de la traduction pour exposer les mécanismes linguistiques sous-tendant la traduction humaine est un luxe pour des étudiants qui doivent déjà maîtriser deux ou trois langues, une ou deux spécialités, les techniques de rédaction et de traduction, la linguistique et ses sous-branches, les techniques de recherche documentaires, etc.

### 3.2. Didactique d'un langage-macro

Comme on l'a vu ci-dessus, l'enseignement d'un langage de macro-commandes est globalement plus aisé que celui d'autres langages de programmation.

Le langage-macro comporte des instructions propres, pour les applications liées au logiciel dont fait partie ce langage. Pour WORD 5.0 : test du numéro de la fenêtre active à l'écran (IF FENETRE=*n*) ; test du contenu de la mémoire-tampon de texte ; test de la réussite d'une recherche de chaîne de caractère ou d'instruction de mise en forme (IF TROUVE) ; test du contenu d'une option de menu ou de sous-menu du programme ; test du contenu de la portion de texte sélectionnée ; test de la présence d'un zoom sur une fenêtre de l'écran ; modulation de l'interaction avec le programme. Les réponses attendues par les commandes de WORD peuvent provenir, au choix de l'utilisateur, de la macro elle-même, de l'utilisateur ou bien ne pas être considérées par le programme. Les versions ultérieures de WORD ont enrichi fortement ce répertoire d'instructions de « traitement de texte ».

Les langages-macros requièrent d'autres compétences spécifiques que celles de leurs instructions propres. La maîtrise des mouvements du curseur, très nombreux dans les applications manipulant du texte (aller-retours du curseur entre menus de commandes et texte ainsi qu'entre fenêtres multiples), requiert l'attention du professeur. En WORD, par exemple, la pression de la touche F10 sélectionne le paragraphe actif ou le paragraphe suivant (si le paragraphe actif avait déjà été sélectionné) ; si, au cours du déroulement du programme, le paragraphe en question se résume à un retour-chariot et que le curseur se trouve sur lui, la pression de F10 le fera choisir le paragraphe suivant et non ce retour-chariot. Autre exemple : la recherche globale d'une chaîne de caractères. Si, au cours du déroulement d'une

---

<sup>10</sup>. elle ne l'est pas en Belgique

macro, le curseur se trouve sur une partie de la chaîne ou sur toute la chaîne à trouver lorsqu'il doit effectuer sa recherche, il ne trouvera bien sûr pas cette occurrence et passera à la suivante. Il peut dès lors s'avérer utile de prévoir dans certains programmes un recul *ad hoc* de deux ou trois caractères pour le curseur (si les occurrences de la chaîne sont presque juste à côté l'une de l'autre). Il arrive même que la première occurrence soit tout au début du document. Dans ce cas, il faut ajouter quelques caractères, faire remonter le curseur en haut du document, lancer la recherche, puis effacer ces lettres « postiches ».

Au début, l'hétérogénéité des langages de macro-commandes est source de confusion pour l'étudiant : dans un seul programme de traitement de texte, l'ordinateur peut en effet faire des choses aussi multiples que « presser » lui-même des touches, choisir des options de menus, charger une macro-commande ou une entrée de glossaire, lier une feuille de style à un document (donc automatiser toute une typographie ou une composition de texte par un autre mode qu'un programme), lancer un sous-programme, ouvrir des fenêtres à l'écran, supprimer l'affichage des menus de WORD lui-même, sortir de WORD pour lancer une commande en DOS puis revenir dans le traitement de texte, ou enfin exécuter des instructions de programmation classiques. Heureusement, les différentes catégories de commandes se distinguent nettement par leur syntaxe : en WORD, les noms de touches sont entre angles (une instruction de pression de la touche retour-chariot ENTER s'écrit *<retour>* ; les instructions de structuration, les messages, les affectations, les noms de variables, à exécution différée (elles ne peuvent donc être générées par enregistrement mais doivent être écrites) sont encadrées de chevrons : *<> ""*), *«comment» ... «endcomment»*, *«while c<10»* (dommage que l'anglais et le français y soient mélangés!) ; les affichages à l'écran, les appels de menus, les choix d'options et les communications à WORD de paramètres s'écrivent comme des chaînes de caractères simples : *<ctrl annule>fpo<retour>* signifie : pression de la combinaison de touches CTRL+ESCAPE pour positionner le curseur dans la zone des menus de WORD, choix du menu FORMAT (frappe de la lettre *f*), choix du sous-menu PARAGRAPHE (*p*), choix de l'option OUI (*o*), pression de ENTER pour lancer la commande de mise en forme (en l'occurrence, la mise en grasses d'une portion de texte préalablement sélectionnée).

### 3.3. Limites des langages-macros

Le langage-macro n'est manifestement pas fait pour des programmes-applications complets : les saturations de mémoire vive y sont la règle : WORD 5.0 n'exploitait pas plus de 720 K ; le nombre de variables est limité (64 par macro pour WORD 5.0). Le dépassement d'autres seuils entraîne également une fin prématurée de programme, par exemple, pour WORD : chiffre supérieur à 32 760 ou inférieur à 0 dans une comparaison ; résultat d'opération excédant 14 chiffres ; macro comportant plus de 16 sous-macros ; imbrication de plus de 3 instructions REPEAT (itérations définies).

Les langages-macros ne sont pas complets et comportent des bogues. En WORD 5.0, le nombre de fonctions est réduit : les seules fonctions de manipulation de chaînes prévues sont le comptage des signes d'une chaîne et le contenu d'une

sous-chaîne mais cette dernière fonction ne fonctionne pas ; la création de fonctions d'utilisateur est en outre impossible. Toujours en WORD 5.0, les conditions multiples ne sont pas prévues, les opérateurs mathématiques disponibles se réduisent aux quatre opérations arithmétiques de base et au pourcentage ; les seuls opérateurs booléens proposés sont le ET, le OU (inclusif) et le NON ; l'instruction GOTO n'existe pas.

### **3.4. Programmes en traitement de texte**

Voici quelques exemples de programmes appliqués à la traduction. Leur brièveté est surprenante, par comparaison à celle des programmes écrits avec des langages de programmation classiques. Facturation ou devis de travaux de traduction, après comptage des lignes ou des mots d'un texte ; chargement d'un glossaire terme à terme dans une fenêtre et remplacement des termes sélectionnés dans la traduction par l'équivalent en langue étrangère, stocké dans ce glossaire ; inversion de glossaires (tri sur l'autre ou sur une autre langue) ; préparation d'un dictionnaire pour l'édition (mise du premier et du dernier mot ou terme de la page en titre courant, dans la marge, présentation en colonnes, typographie des vedettes et des renvois, choix du format de page, etc.) ; assistance à la création de dictionnaires (génération automatique de renvois inverses et de l'index des mots des différentes langues, etc.) ; calculette, convertissant des valeurs monétaires, des mesures etc., d'un système national à un autre ; gestion des archives de traduction ; conversion d'un document de format d'un traitement de texte pour son utilisation par un autre traitement de texte (celui du client du traducteur) ; corrections stylistiques .

### **3.5. La fin des langages-macros ?**

Le remplacement du langage de macro-commandes par le QUICKBASIC, dans les versions de WORD pour WINDOWS, est dû à la sous-utilisation des langages-macros par le grand public. Peut-être l'enseignement se prépare-t-il trop tard à jouer son rôle à cet égard ? Ces langages sont, comme on l'a vu, insuffisants pour les programmeurs professionnels. Cette substitution s'inscrit dans la tendance à diviser les tâches : convivialité WINDOWS pour les utilisateurs, rendant les manuels d'utilisateurs quasiment superflus et automatisation des applications par les professionnels grâce à un vrai langage de programmation.

Ce remplacement est dommage : même si ce langage de programmation permet toujours les enregistrements de commandes, il allonge l'enseignement de la programmation. A titre indicatif, à l'Institut Marie Haps, le cours de programmation en WORD 5.0 prenait 15 heures/année, pour 30 heures en DBASE III Plus (avec des performances des étudiants nettement moins bonnes).

**Caroline de SCHAETZEN**

Institut Libre Marie Haps  
Chargée du cours d'outils informatisés  
d'aide à la traduction