

informatique de nos étudiants sont inversement proportionnelles à la diffusion des micro-ordinateurs et à la convivialité de leurs logiciels et interfaces. Une série de cadres d'entreprise ignorent eux aussi dans quel logiciel ils travaillent quotidiennement. Certains ne savent pas non plus qu'ils utilisent jusqu'à trois logiciels différents pour une activité. Selon nous, cette évolution est partiellement bénéfique parce qu'elle permet la concentration sur les tâches professionnelles. Une condition s'impose pour qu'elle le reste: le service offert par les fabricants d'ordinateurs et de logiciels doit s'améliorer et être disponible à toute heure. Les problèmes nécessitant des connaissances en informatique générale n'ont en effet pas disparu.¹

La titulaire d'un cours de logiciels verticaux (professionnels) aux traducteurs, interprètes et terminographes que nous sommes aimerait être déchargée de l'enseignement des logiciels horizontaux. Utilisés dans une série de professions de bureau, systèmes d'exploitation, interfaces graphiques, traitements de texte, tableurs, gestionnaires de bases de données, gestionnaires de fax, etc. ne pourraient-ils faire l'objet d'un enseignement à tous les élèves du secondaire? Même s'il était élémentaire, le programme du supérieur s'en trouverait allégé. Le statut optionnel du cours d'informatique complique la mise à niveau d'une population étudiante.

La formation en informatique professionnelle au sens large répond à une demande claire des étudiants, des professionnels et des candidats à ces professions. Jusqu'à quel point les recherches des chômeurs seront-elles facilitées par le petit bagage en bureautique acquis à l'Institut? Nous l'ignorons. A en croire la proportion stable de personnes sans ordinateur venues à l'Institut par le bouche à oreille, le premier contact avec un PC par le truchement d'un tutoriel semble positif. Aucun échec de manipulation, même petit, ne se produit avec eux et l'apprentissage peut se faire au rythme de l'apprenant.

Chez nos traducteurs-fonctionnaires, le parc du matériel professionnel semble plus récent que les logiciels. Tordons également le cou à la légende selon laquelle l'enseignement supérieur est quelque peu coupé des réalités professionnelles. Actuellement, les étudiants en traduction de notre Institut connaissent plus d'outils et des outils plus récents que ceux d'une partie de la profession belge. Environ un traducteur ou chef de service de traduction par mois nous demande (« vous qui êtes neutre ») un avis de type « Magazine de consommateurs » sur tel produit ou tel type de produit. La réponse à ces demandes n'est pas facile. Elle requiert le test permanent, donc l'achat, de tous les produits du marché et des centres de recherche. Elle relève dès lors du travail de consultant et requiert un budget trop important.

Les recyclages offrent des bénéfices secondaires à ceux qui les suivent. Au fil de leur recyclage en informatique, certains traducteurs en exercice ont été revalorisés. Leur raisonnement -informulé- était le suivant: une profession qui fait l'objet de tant de programmation est aussi importante que les professions

informatisées depuis plus longtemps, celles du chiffre au sens large. De plus, les traducteurs rencontrent des collègues aux recyclages. En Belgique, les traducteurs des ministères et para-stataux sont éparpillés, pour être proches de leurs maîtres d'ouvrage, et certains ne peuvent téléphoner librement à l'extérieur. Les discussions avec leurs homologues ont donc développé leur esprit de corps. A long terme, le battage médiatique autour d'Internet a quant à lui le mérite d'attirer l'attention du grand public sur le coût des errances lors des recherches d'informations. Il soulignera le rôle des techniques de recherche documentaire, et, plus globalement, de cet « or gris » qu'est l'information unilingue ou multilingue.

C. de SCHAEZTEN et Th. LEPAGE
Institut Marie Haps, rue d'Arlon,
111040 Bruxelles- Belgique
Tél.: (02) 511 92 92 Fax: (02) 511 98 37

L'Enseignement des Logiciels Professionnels: Spécificités et Contraintes

Caroline de SCHAETZEN

Les techniques d'enquêtes posent enfin une question accessoire. Le sexe doit-il garder le statut de variable de contrôle qui lui est classiquement réservé dans les questionnaires?

Toile de fond

Les logiciels verticaux sont des logiciels destinés à un métier ou une discipline tandis que les logiciels horizontaux informatisent des tâches communes à plusieurs d'entre elles, comme l'écriture (traitement de texte), le calcul (tableurs), le traitement de fichiers (systèmes de gestion de bases de données), la communication (logiciels de gestion de télécopies)... L'Institut Libre Marie Haps forme depuis plus de treize ans à des logiciels verticaux, à savoir ceux de traduction, de lexicographie et de terminographie assistées par ordinateur. Le public suivant cet enseignement et celui de certains logiciels horizontaux est le suivant:

- Etudiants en traduction et en interprétation (niveau d'étude: Bac +2), pour les cours de traduction et d'interprétation assistées par ordinateur.
- Traducteurs en exercice et, à partir de l'année académique prochaine, interprètes en exercice, pour les cours de traduction et d'interprétation assistées par ordinateur.
- Terminographes en exercice, c'est-à-dire auteurs de dictionnaires unilingues, bilingues ou multilingues de langue technique ou scientifique
- Lexicographes en exercice, c'est-à-dire auteurs de dictionnaires unilingues, bilingues ou multilingues de langue générale, pour les cours de terminographie ou de lexicographie assistée par ordinateur.
- Professeurs de traduction, pour les cours de traduction assistée par ordinateur.
- Professeurs de terminographie et de lexicographie, pour les cours de terminographie ou de lexicographie assistée par ordinateur.
- Professeurs de l'enseignement secondaire, pour le cours de traitement de texte. J.-L. Baron (1990) a en effet montré que les logiciels horizontaux pouvaient servir dans l'enseignement de la plupart des disciplines ainsi que pour toute la bureautique professorale.
- Secrétaires de l'Institut, pour le tutoriel d'initiation au traitement de texte ainsi que pour un tutoriel d'initiation à Dos et à Windows.
- Grand public, surtout des chômeurs, pour le tutoriel d'initiation au traitement de texte, ainsi que pour un tutoriel d'initiation à Dos, un tutoriel d'initiation à Windows et un tutoriel de micro-informatique générale. J.-L. Baron (*op.cit.*) a fait remarquer que les logiciels horizontaux « et les concepts qu'ils mettent en oeuvre s'intègrent progressivement dans la culture générale ».

La présente contribution tente d'abstraire de cette expérience quelques généralités sur les contraintes et les spécificités de l'enseignement des logiciels verticaux et horizontaux utilisés pour l'automatisation d'un métier précis.

Les différences d'âge et leur incidence sur la motricité, les divergences dans le « vécu » scolaire et éventuellement universitaire, dans la « culture informatique », les connaissances théoriques et pratiques, les capacités d'apprentissage, ne s'observent pas dans une classe d'élèves ou d'étudiants. Si cette réalité n'est pas prise en compte par des mises à niveau préliminaires ou la constitution de groupes, un cours-recyclage aux professionnels mécontente tous les participants, forts et faibles. Nous ne nous attarderons cependant pas sur ces spécificités de la pédagogie pour les adultes, largement traitées dans la littérature. Les besoins des apprenants adultes ont par exemple fait l'objet de typologies, D. Beau (1983) distinguant par exemple les connaissances insuffisantes, connaissances importantes mais vieilles, aptitudes insuffisantes, attitudes inadaptées.

PARC DIDACTIQUE

« La question des moyens, parfaitement dédaignée par les budgétaires et mal pensée par les administratifs », dit L. Sfez (1991), sous-tend la formation à l'informatique d'un métier. Pour fixer les idées, les paragraphes qui suivent détaillent le parc informatique requis dans une faculté ou un Institut pour traducteurs et interprètes. Le matériel et les logiciels des établissements d'enseignement supérieur de type « Fachschulen » allemandes doit épouser ceux des professionnels. Il contraste avec celui de la recherche, par exemple en informatique linguistique, constitué d'un mini-ordinateur de type « Sun » et de terminaux.

Prix du parc

Tous les prix sont donnés à la pièce et en francs belges (multiplier par 6 pour avoir l'équivalent en francs français). Ils datent de juin 1996.

Prix des logiciels

La liste des logiciels reprend ceux qui sont indispensables au traducteur (sauf le logiciel de traduction automatique, destiné à un service ou une agence de traduction). Elle comporte une seule marque, la plus vendue, par type de logiciel.

Pour chaque logiciel, le tableau précise les caractéristiques-plancher requises pour le matériel. Mabiliothèque tourne par exemple sous Dos, donc le tableau n'indique pas qu'il peut être lancé depuis Windows. Par contre, lorsque le tableau indique une version Dos et une version Windows, c'est qu'il en existe effectivement une pour chaque plateforme.

<i>Logiciel</i>	<i>Prix</i>		<i>Matériel requis</i>
	<i>Version Dos / Unix</i>	<i>Version Windows</i>	
Traitement de texte Word 6 (il faut acheter séparément le vérificateur pour chaque langue utilisée dans les rédactions. Il n'existe pas de vérificateur syntaxique pour d'autres langues que le français et l'anglais)	16.000/6.000 (mise à jour) + 6.000 (vérificateur d'orthographe)	16.000/6.000 (mise à jour) + 6.000 (vérificateur d'orthographe)	A installer sur disque dur Pour Dos: processeur 286 Pour Windows: processeur 386, RAM 8 Méga, espace-disque 20 Méga (installation complète, vérificateur orthographico-grammatical pour 2 langues)
Dictionnaire sextilingue de synonymes Lexica pour la la langue courante	1.500		A installer sur disque dur 8088, RAM 640 K, espace-disque 2 à 4 Méga (selon nombre de langues installées), Dos 2.11, disquette 3 ou 5 pouces
Dictionnaire Larousse anglais-français et vice-versa pour la langue générale	3.500	3.500	Sur disquette ou CD-ROM Pour Dos: RAM 8 K, espace-disque 2 Méga, Dos 3.0 Pour Windows: RAM 120 K, espace-disque 2 Méga, Dos 3.0, Windows 3.0
Banque décalingue de termes EURODICAUTOM	gratuite		A installer sur disque dur 8088, modem
Gestionnaire Easyfax de fax et de télécommunication de données	gratuit (livré avec le modem)	gratuit (livré avec le modem)	8088, modem
Exploiteur d'archives de traductions Translation manager		100.000	386, RAM 8 Méga, espace-disque 15 Méga, Windows 3.1
Traducteur automatique METAL néerlandais-français et vice-versa	1.000.000		A installer sur disque dur Unix, Mini-ordinateur Sun, RAM 32 Méga
Lecteur optique multilingue + stylo Datapen/ Logiciel de lecture optique pour lecteur de table (13 langues, dont le russe)		12.000/ 26.000	386, RAM 4 Méga, espace-disque 5 Méga, Windows 3.1
Gestionnaire de documentation individuelle Mabiliothèque	500		Sur disquette ou à installer sur disque dur 8088

Prix du matériel

La multiplication des dictionnaires sur CD-ROM éliminait de notre liste les PC dépourvus de lecteur de CD: un éditeur français a mis sur ce support ses quarante dictionnaires spécialisés anglais-français et sa collection s'enrichit en permanence.

La liste ci-dessous est donnée pour mémoire. Elle inclut les composantes qui prolongent la vie d'un micro-ordinateur. On notera que si le disque dur des PC est relativement grand, leur mémoire vive est sous-dimensionnée pour les applications textuelles. Nous n'avons pas repris le matériel pédagogique proprement dit, comme un réseau didactique et un projecteur-agrandisseur de contenu d'écrans.

<i>Composante</i>	<i>Caractéristiques</i>	<i>Prix</i>
Micro-ordinateur de bas de gamme	Pentium à 175 Mhz, 16 Méga RAM, disque rigide de 1 Giga, écran 17 pouces SVGA couleurs, une unité de CD-ROM (Windows 95 fourni gratuitement)	45.000/ 70.000
Modem de bas de gamme	Externe, 9000 pi, 27000 Bauds	3.500 /7.000
Lecteur de CD-ROM	Quadruple vitesse, avec driver et carte de son	10.000
Disque rigide	540 Méga, avec driver	9.000
Barrettes de supplément de mémoire RAM	Par module de 4 Méga	8.000
Carte-mère 486 pour 286	Carte-mère 486	10.400
Imprimante de bas de gamme	Laser de table, résolution 600 caractères par pouce, 8-12 pages par minute, 2 Méga de RAM	30.000/ 45.000

Des économies

En raison du marché que représentent les étudiants et les professionnels en recyclage, un accord avec les fournisseurs de logiciels est souvent négociable. Ceux-ci accorderont par exemple une version pour réseau des logiciels professionnels au prix d'une version monoposte.

Avec un peu de collaboration, une rotation du matériel est praticable. Les PC d'avant- ou avant-avant-dernière génération peuvent être cédés à l'administration de l'établissement d'enseignement. Ils peuvent ensuite revenir dans la salle d'informatique didactique quand ils sont déclassés, par exemple pour faire tourner des tutoriels peu gourmands en ressources.

A quand l'organisation par les ministères de l'éducation d'un partenariat systématique avec les entreprises pour faire bénéficier les étudiants de leur matériel déclassé?

L'écart de prix entre les produits blancs et les gammes bon marché des PC de marque s'amenuise. Les PC sans marque ne présentent pas les avantages offerts par les grands fabricants: sponsorship de colloques, prêt de matériel pour certaines manifestations, livraison d'une série de logiciels avec l'ordinateur. Leur qualité peut laisser à désirer, comme celle des haut-parleurs des ordinateurs multimédia. Le service après-vente est inexistant pour eux, un inconvénient pour des parcs importants. A condition qu'il soit assez vaste, les gros vendeurs d'ordinateurs peuvent reprendre le parc déclassé des grandes marques pour le revendre dans un pays en voie de développement. Par contre, les produits blancs sont sans grande valeur sur le marché d'occasion.

Notons que le budget en livres et en articles du professeur d'informatique professionnelle est quatre fois plus élevé parce que plus obsolescent que celui des enseignants exposant d'autres volets du métier.

Ecoute du marché

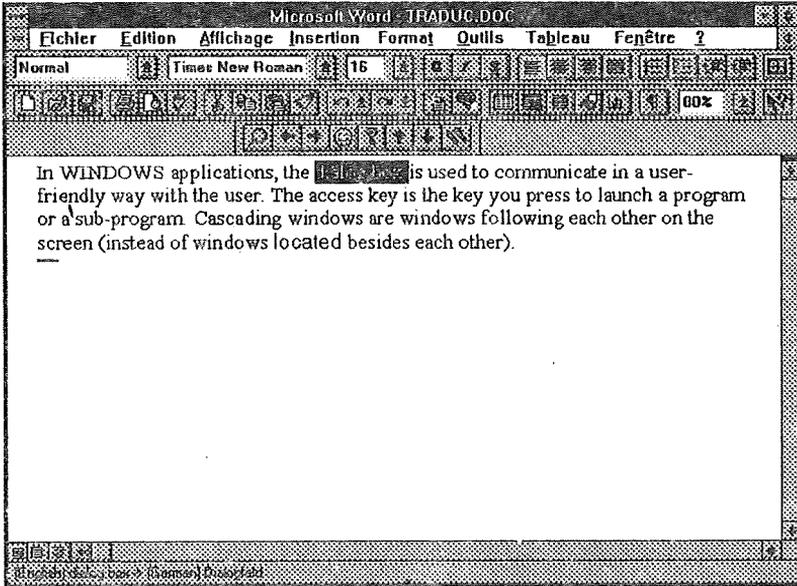
Si le responsable du parc didactique n'anticipe pas l'évolution du marché, il se retrouve parfois en possession de logiciels et de matériel inutilisables après un ou deux ans. Cette prévision n'est pas toujours aisée même si une règle se dégage: le primat de la diffusion et des standard. Le logiciel est devenu une marchandise comme les autres et c'est souvent le plus vendu qu'il faut acheter, même dans l'enseignement.

Un premier exemple de prévision difficile est la mauvaise qualité des traducteurs automatiques pour micro-ordinateurs, qui en fait des produits inutilisables. Par contre, le serveur Compuserve utilise Logos, un système de traduction automatique très ancien -et défaillant- pour traduire toutes les trois minutes les dialogues d'un groupe de conversation Internet sur la photographie d'amateurs. Le succès de ce service très bon marché a permis à Compuserve de financer à lui seul la programmation d'une nouvelle version de Logos. Cette version répondra aux standards de la linguistique informatique d'aujourd'hui, donc requerra pourtant une grosse équipe de développeurs, de linguistes et de traducteurs. Or d'habitude, c'étaient des universités que provenaient les meilleurs programmes de traduction automatique.

Une seconde évolution difficile à prévoir en Belgique fut le fait que Word n'a supplanté WordPerfect que depuis leurs versions respectives pour Windows. Une question subsidiaire se pose. Les logiciels de traitement de texte multilingues et multi-alphabet n'ont jamais eu de succès chez les professionnels du multilinguisme, mais l'actuelle concentration du marché autour de Word et WordPerfect laissera peut-être plus de place à Accent, un traitement de texte multilingue israélien dont le marketing est bien orchestré. Devons-nous l'acheter ou nous restera-t-il sur les bras comme les trois produits multilingues pour Dos, acquis il y a 5 ans? N'avons-nous de toutes manières pas un rôle à jouer dans la diffusion de ce type de produit spécialisé?

Le nouveau rôle-pivot du traitement de texte pour d'autres logiciels manipulant du texte est une autre évolution surprenante. Les logiciels de traduction automatique, notamment, possédaient leur module de traitement de texte, particulièrement adapté aux révisions des textes traduits. A présent, les auteurs de ces logiciels déploient de gros efforts pour s'interfacer très étroitement avec Word, WordPerfect, AmiPro. Cette interface détermina également le succès commercial de l'exploiteur de traductions antérieures programmé par une petite société allemande. Cette préférence s'est marquée aux dépens du Translation Manager, premier logiciel à exploiter automatiquement les textes traduits (et consultant automatiquement les glossaires « clients » du traducteur) et vendu par IBM. Mais les traducteurs n'ont pas le courage d'apprendre le module de traitement de texte de

Translation Manager... Voici un exemple de la qualité de l'interface d'un gestionnaire de glossaires, c'est-à-dire un logiciel facilitant la confection et la consultation de petits dictionnaires multilingues conçus par les traducteurs au fil de leurs traductions (le « jargon » de Renault diffère de celui de Ford, d'Opel et peut même différer d'un modèle de voiture à l'autre, chez un même constructeur de voitures):



Le traducteur a mis en surbrillance le terme anglais et cliqué sur l'icône de la loupe. Trouvé instantanément dans le glossaire ouvert en arrière-plan, l'équivalent allemand de *dialog box* figure dans la barre d'état de Word 6. Un clic! sur l'icône du « smiley » l'insérerait dans le texte à traduire. Une seconde interface de ce gestionnaire de glossaires a été stockée par ses auteurs dans une macro-commande programmée en WORDBASIC. Cette macro génère en Word 6, WordPerfect ou AmiPro une version prête à cliquer du glossaire ouvert en arrière-plan. En voici un extrait:



L'enseignement du traitement de texte dans le secondaire revêt donc une importance accrue. S'il se généralise, les formateurs des diffuseurs que sont les traducteurs, les journalistes, les avocats, les secrétaires, les rédacteurs et les écrivains, pourront se concentrer sur l'enseignement des seules fonctions avancées du traitement de texte, à l'aide d'exemples professionnels (un formulaire sert aux traductions de contrats, etc.)

options pédagogiques

Les cours préparant ou recyclant au volet informatique d'une profession posent des questions pédagogiques propres.

Combat ou utilisation des représentations?

« Les représentations sont des îlots de connaissance, sans liens cohérents. Elles dépendent de la subjectivité des individus. Elles résistent au changement. Elles sont surtout faites d'images mentales: c'est la traduction en termes généraux d'une perception partielle de la réalité », dit D. Beau (1983).

L'optique informatique du gestionnaire des documents entre en conflit avec la conception que se fait l'apprenant d'un texte. Elle date de la machine à écrire alors que, pour un traitement de texte, un document est « une suite de caractères; c'est une suite de paragraphes; c'est une suite de pages; c'est une suite de sections ou de divisions (...) », dit J.-Fr. Lévy (1995). De même, « pour l'utilisateur, la notion d'activation de fenêtre n'a pas vraiment son équivalent dans le domaine matériel connu ». « Certains termes, même s'ils représentent un progrès (*ouvrir* au lieu de *charger*, par exemple), n'ôtent pas toute la complexité des opérations qu'ils commandent et donc des concepts sous-jacents; cette facilitation déplace les problèmes des représentations des objets manipulés par l'informatique, sans les supprimer », ajoute-t-il.

Les représentations se nichent là où on les attend le moins. Le piratage des logiciels est un des grands responsables de la lenteur de diffusion des logiciels professionnels, dont le nombre d'exemplaires est moins grand que celui des logiciels horizontaux, donc le piratage, plus facile à déceler. La copie du traitement de texte est la règle et ce, par principe: on investit dans le matériel mais jamais dans les logiciels. Certains traducteurs ont troqué leur 486 pour un Pentium mais font et consultent encore leurs glossaires avec leur traitement de texte ou -pire!-, avec un tableur... Une campagne sur le capital intellectuel que représentent la programmation et l'analyse s'impose avant tout enseignement de logiciels professionnels.

Les attentes diffuses des étudiants à l'égard du cours d'informatique contrastent quant à elles avec le caractère de plus en plus ciblé des logiciels professionnels. Le caractère tout aussi dédié des logiciels de bureautique n'est pas perçu des apprenants, surtout les étudiants. Le formateur doit prendre attitude face à ces images d'aspinal d'une informatique universelle, qui ont la vie dure. A l'issue de leur formation, plusieurs licenciés en traduction-interprétation de notre Institut se sont par exemple inscrits à des cours qu'ils qualifiaient de « vraie » informatique, entendez, une formation à Excel, dBase ou à Visual Basic. La formation dispensée par IBM les éloignait d'autant plus de leur métier qu'elle portait sur un système d'exploitation et un langage de programmation idiosyncrasiques à un *mainframe*. Une parade possible du professeur est la théâtralisation. Au début de l'année académique, nous consacrons désormais deux heures à préparer les étudiants au type de logiciels qui leur seront montrés. Comme en judo, l'énergie du mouvement adverse peut être captée et détournée de son but: bien orchestrés, l'enseignement des outils professionnels, le ciblage des démonstrations et exercices renforcent les motivations des étudiants pour la traduction et l'interprétation elles-mêmes. Les associations d'anciens étudiants sont également un bon canal de rétroactions.

L'effet « Bill Gates » n'épargne pas l'enseignement supérieur: certaines facultés de lettres enseignent... un tableur à leurs étudiants. Pourtant, une trentaine de logiciels assistent la traduction, l'interprétation, la terminologie et la lexicologie. Les débouchés accessibles aux diplômés en lettres sont plus nombreux que ces quatre métiers; ils comptent les professions de l'édition et l'enseignement des langues. Pour ce dernier, la palette de didacticiels, d'exerciceurs, de langages de programmation didactiques, de systèmes d'auteurs, est immense... Pour l'analyse littéraire, le concordancier extrait le vocabulaire des textes sur support électroniques et dresse des statistiques lexicales. Ce générateur de concordances (listes alphabétiques de mots avec leur contexte) existe depuis les années cinquante. Sans compter toutes les applications d'ingénierie linguistique, ou au moins ses noyaux polyvalents comme les analyseurs syntaxiques et les générateurs de textes... Malgré le ciblage sur la profession de nos cours d'informatique, les professeurs de notre propre établissement relayent les représentations véhiculées par les médias. nos logiciels de traduction assistée les ont souvent laissés de marbre mais ils ont demandé un accès à Internet. Or, pour les traducteurs, ce dernier n'est qu'une voie

d'accès à de la documentation spécialisée primaire (livres, articles, rapports...) et secondaire (dictionnaires).

Exploration ou exercices dirigés?

Une priorité doit être établie entre l'apprentissage actif, exploratoire, et l'aperçu complet des potentialités d'un outil pour les tâches professionnelles. L'un exclut malheureusement l'autre lorsque le nombre d'heures de cours est limité.

Le sous-emploi des logiciels est fréquent chez les professionnels. Montrer aux apprenants les opérations déjà automatisées nous paraît donc prioritaire. Ils sauront ainsi que telle tâche a fait l'objet d'une commande ou d'un logiciel, quitte à ce qu'ils en cherchent le mode d'accès dans un manuel. Sans cela, comment éviter les bricolages comme, en traitement de texte, le centrage horizontal d'un titre par des espaces ou le remplacement des tableaux par des tabulations, pour ne citer que les plus flagrants? Les enquêtes menées chez des traducteurs indépendants et fonctionnaires nous ont montré que ces utilisateurs ne lisent pas leurs manuels. Ils se bornent à consulter épisodiquement l'aide en ligne des logiciels, ou à procéder par essais et erreurs. Les utilisateurs ignorent des commandes mais aussi leur résistent. Les futurs professionnels du verbe ayant confiance en leurs connaissances linguistiques, il nous a par exemple fallu prouver l'utilité du vérificateur syntaxique. Nous avons rappelé le statut professionnel de la relecture orthographique-syntaxique: toutes les maisons d'édition emploient à temps complet ou partiel une batterie de correcteurs d'épreuves.

On ne peut faire découvrir un grand nombre de commandes de logiciels. L'exploration sur base des seuls manuels d'utilisateurs, tous différents et de qualité inégale, prend trop de temps. Les « découvertes orientées » sur ordinateurs avec corrigé font voir ne fût-ce qu'une fois toutes les commandes accessibles et ce, avec l'usage qui peut en être fait pour un métier. Voici un exemple de ces exercices:

Demandez de voir les fiches bibliographiques des livres portant uniquement sur la terminologie

- 1) Pressez ENTER pour choisir l'option Liste des livres.
- 2) Descendez avec la touche « flèche vers le bas » jusqu'à Mots-clés et dactylographiez terminologie. Pressez ENTER
- 3) Répondez aux deux questions posées au bas de l'écran en pressant deux fois la touche o et pressez la touche 1 pour faire afficher les fiches sélectionnées.
- 4) Feuillotez les références bibliographiques à l'aide des touches PGUP et PGDN.
- 5) Pressez F3 pour revenir au menu principal.

Moins passif que des démonstrations, cet encadrement ménage une plage horaire à l'exposé des nouvelles tâches, certains logiciels avancés ou très complets réorganisant des pans entiers d'une profession.

Séparation des acquisitions ou pédagogie par tâches?

Un autre choix pédagogique s'opérera entre l'intégration des apprentissages et partant, l'assurance du transfert des acquis, et la séparation des acquisitions. L'alternative est le cours de la discipline ou du métier donné sur ordinateur ou le cours séparé sur les logiciels. L'informatisation des cours d'une discipline ou d'un métier (dans notre cas, ceux de traduction, d'interprétation, de lexicographie et de terminographie) émule l'environnement professionnel (pour nous, la transposition linguistique orale et écrite « bureautisée »). Comme les professeurs ne sont pas habitués à donner cours avec un collègue, que ce soit le professeur d'informatique ou un autre collègue de la discipline, leur formation à tout ou partie des logiciels s'impose. L'informatisation des cours requiert en outre une coordination entre les professeurs (de traduction, par exemple), pour se répartir les logiciels à utiliser pendant leurs cours. De plus et surtout, les cours sur machine requièrent une multiplication du nombre d'ordinateurs. Enfin, chez les adultes, les transferts d'apprentissages sont plus rapides que chez les enfants et les adolescents parce que les apprenants peuvent y être sensibilisés. Un des pires effets secondaires du cloisonnement des enseignements disparaît donc.

L'informatisation des cours est indispensable mais peut s'étaler sur plusieurs années. Actuellement, les professeurs de traduction et d'interprétation de notre Institut demandent de temps à autre des exercices d'informatique traductive aux étudiants: traduction chronométrée sur machine, avec devis et reproduction fidèle de la mise en page de l'original. Un professeur de traduction espagnole et un professeur de traduction russe donnent un cours sur PC toute l'année.

Prestige ou pragmatisme?

Spécialiste de la discipline ou de la technique enseignée ou de l'informatique elle-même, le professeur universitaire est tenté de proposer aux étudiants ses propres programmes, même s'ils ne représentent pas l'état de la recherche ou des produits du marché. Un établissement belge enseignant la traduction fait utiliser un gestionnaire de glossaires qu'il avait programmé mais qui n'a jamais pénétré chez les traducteurs parce que son alimentation était bien trop lourde.

« L'acte de formation réussi entraîne chez le formé un changement et ce changement est observable », disent R. Besnaiou, C. Muller et Ch. Thouin (1988). A l'université et dans le supérieur non-universitaire, le prestige du professeur sera bien plus grand s'il enseigne la programmation que des logiciels professionnels. Or vu leur spécialisation et leur nombre, seuls les seconds rendent l'apprenant directement opérationnel.

Pour le professeur d'informatique, les cours magistraux sont plus gratifiants que le statut de facilitateur d'accès au savoir. Le responsable d'un laboratoire d'informatique est en effet perçu peu ou prou comme un bibliothécaire. Aussi, les professeurs d'informatique de l'enseignement supérieur n'adopteront-ils sans doute ce rôle que contraints et forcés. Généralement, ce sont des assistants qui se voient

confier les centres de ressources en informatique. Une dimension des logiciels signalée par P. Lévy ne leur est pas perceptible: « la plupart des logiciels contemporains jouent un rôle de technologie intellectuelle: ils réorganisent peu ou prou la vision du monde de leurs utilisateurs et modifient leurs réflexes mentaux ».

Formation ou informations?

Les attentes des professionnels sont précises. Le ciblage des exercices et des démonstrations sur la profession est dès lors impératif. Il facilite leur réflexion sur la réorganisation de leur travail autour des outils. Mais comme l'environnement de travail reste dans leur esprit, leurs attentes dépassent les cours proprement dits.

Les professionnels souhaitent tout d'abord des études de marché détaillées pour les logiciels, assortie de comparaisons de type « Test achats (la revue de l'association belge des consommateurs). Or pour l'établissement dispensateur de la formation, des analyses de ce type sont chères et longues à mener. On demande consultants bon marché pour la micro-informatique professionnelle...

Les formations continuées incluent des publications parce que les professionnels demandent également des informations sur les usages des nouveaux types de logiciels apparus sur le marché. L'organe de leurs associations professionnelles est le plus sûr moyen de les leur procurer. Nous publions donc dans les revues de traduction et de terminologie une description des fonctions et des applications de tous les outils dont nous avons connaissance. En dix ans, le nombre de ces articles didactiques dépassera la centaine. R. Lelouche (1992) était d'avis que « dans un contexte de formation continue, l'enseignant doit avoir une double formation. D'une part, il doit avoir une solide formation de base en informatique, afin d'éviter que son cours ne devienne une suite de recettes. Mais il doit aussi avoir une connaissance effective des activités ou de la discipline ». On peut se demander si la formation en informatique n'est pas secondaire par rapport à celle de la discipline. Comment un informaticien jugera-t-il des performances d'un logiciel de traduction automatique, les critères étant devenus uniquement linguistiques?

Un professionnel est tributaire d'un ou plusieurs supérieurs hiérarchiques pour le choix de ses logiciels. Un coup de pouce est dès lors parfois demandé au professeur lorsque l'apprenant envisage l'acquisition d'outils. Un traducteur de la Fonction publique désireux de s'opposer à l'installation de dBase parce qu'il lui préférerait tout naturellement un gestionnaire de glossaires multilingues nous a par exemple demandé des arguments pour convaincre les décideurs du Ministère. Pour unifier le parc des logiciels et faciliter les formations des agents, dBase avait été choisi pour tous les fonctionnaires. Au sein du gouvernement fédéral belge, certains logiciels horizontaux sont même choisis à l'échelon de plusieurs ministères, ce qui entrave l'acquisition d'outils ciblés pour les professions faiblement représentées comme la traduction... Notons qu'à ces impératifs rationnels des responsables ministériels s'ajoute un manque de considération pour l'ingénierie linguistique en général et pour l'informatique linguistique en particulier. Pourtant, l'engouement

pour Internet montre à suffisance l'importance économique de cet « or gris » qu'est l'information...

Les connaissances du matériel et de l'informatique-système du responsable de l'enseignement des logiciels sont bien sûr également sollicitées. Le professeur est le « sujet supposé savoir », selon l'épithète que J. Lacan attribuait au psychanalyste. Il est perçu par les étudiants comme une « hot line » au moment où ils achètent leur PC individuel. Cet achat est de plus en plus précoce: dans notre Institut, il se produit au cours de la seconde des quatre années d'étude et non plus, comme il y a cinq ans, au sortir de ces études. Lors des mémoires et travaux de fin d'année, des étudiants téléphonent en outre parce que « mon imprimante imprime des lignes verticales noires depuis que j'ai installé la nouvelle version de WORD ». Certains étudiants apportent chez le professeur-docteur la caisse de leur ordinateur ou celui-ci peut dicter des manipulations par téléphone, en leur demandant de lire les messages de l'écran. Pour les professeurs non-informaticiens, certains constructeurs organisent pour les vendeurs une bonne formation générale, qui leur permette d'assurer au service après-vente efficace. Ils ouvrent ces cours à des personnes de l'extérieur. Mais en Belgique, l'interlocuteur rêvé des enseignants est le CeFIS parce que ses cours sont déjà axés sur la gestion de parcs didactiques.

Ces services du professeur ou de son assistant ne font pas l'économie d'un cours de micro-informatique générale, inévitable pour les professionnels et les étudiants tant que le secondaire n'assurera pas une « alphabétisation » à l'informatique. Un cours magistral peut être remplacé par des notes de cours détaillées, qui présentent l'avantage d'être complètes et de ne pas ralentir le rythme de l'exposé. Ces deux dernières années, nous avons largement puisé dans la collection illustrée de Dunod: *La programmation... comment ça marche? Le micro... comment ça marche? Les réseaux... comment ça marche? Les logiciels micro... comment ça marche? Les micro-processeurs... comment ça marche? Le multimédia... comment ça marche? Windows... comment ça marche? La PAO... comment ça marche?* Contrairement à ce que ses titres donnent à penser, cette série destinée au grand public, agréablement illustrée, est à la fois technique et claire. Publiée en français et en anglais, elle peut enseigner dans les deux langues la terminologie de l'informatique élémentaire. Ce bilinguisme en informatique générale est utile à Bruxelles, où la version des logiciels acquis par les sociétés et les ministères est souvent anglaise. La version anglaise sort d'abord, elle est moins chère et l'anglais langue ne heurte pas les susceptibilités linguistiques respectives des francophones et des néerlandophones.

Connaissances ou métaconnaissances?

En raison de la rapide évolution des logiciels professionnels, la matière enseignée inclut les méthodes d'évaluation de logiciels. A titre d'exemple, voici quelques rubriques générales de la grille d'évaluation que nous avons conçue pour les banques de termes, sur base de la grille d'évaluation des logiciels documentaires d'un consultant (Bureau van Dijk, Bruxelles):

1. MACRO-EVALUATION DES BANQUES DE TERMES

- 1.1 Définition des objectifs
- 1.2 Enregistrement des activités
 - 1.2.1 Acquisitions
 - 1.2.2 Notions
 - 1.2.3 Références
- 1.3 Calcul des coûts et des prix de revient
 - 1.3.1 Activités
 - 1.3.2 Entrées (ou enregistrements)
 - 1.3.3 Sorties ou exploitations
 - 1.3.4 Nature des frais
 - 1.3.5 Coûts directs du système
 - 1.3.6 Coûts indirects du système
 - 1.3.7 Coûts indirects de l'organisme ou frais généraux
 - 1.3.8 Grilles de ventilation
 - 1.3.81 Ventilation des coûts indirects sur les coûts directs
 - 1.3.82 Ventilation des coûts d'entrée sur les coûts de sortie
 - 1.3.83 Calcul des prix de revient
- 1.4 Évaluation de l'efficacité
 - 1.4.1 Exhaustivité
 - 1.4.2 Pertinence
 - 1.4.3 Délais
 - 1.4.4 Satisfaction de l'utilisateur
- 1.5 Évaluation de l'efficience
- 1.6 Valeur de la banque
- 1.7 Erreurs fréquentes et améliorations

2. MICRO-EVALUATION DES BANQUES DE TERMES (...)

Le réflexe d'une recherche systématique d'un logiciel professionnel ou des commandes gagne également à être enseigné. Notons que la connaissance de la programmation peut occulter l'existence de logiciels ou de commandes. Qu'il soit professionnel ou étudiant, l'apprenant doit être initié aux circuits documentaires de son professeur lui-même: colloques et collections d'ouvrages dans lesquels sont consignés l'arrière-plan théorique et les résultats des recherches menées dans l'informatique de son métier; collections d'ouvrages et revues annonçant les mises sur le marché des logiciels commerciaux; colloques où sont montrés ces logiciels; revues testant ces produits et conseillant des tours de mains dans leur utilisation professionnelle, etc. L'évolution trépidante du marché impose un volet documentaire à l'enseignement, d'autant plus que les renseignements de première main se cachent souvent dans des sources insoupçonnées ou voyagent.

Le vocabulaire de l'informatique ne devrait pas être gommé des exposés et notes de cours. Indépendamment de son rôle d'ancrage des concepts et si elle est bien définie, la terminologie signale un niveau de technicité et elle est esthétique. La suppression des termes typographiques dans les traitements de texte est regrettable, alors que le savoir-faire des typographes y est réintroduit progressivement. Elle n'est pas systématique et compromet donc la cohérence des vocables (*petites capitales* y voisine avec *majuscules*, au lieu de *grandes capitales*).

Certains équilibres entre consultation et apprentissage doivent faire l'objet d'un enseignement eux aussi. Des traducteurs et des secrétaires qui ont le traitement de texte « dans les doigts » ne gagneront rien à apprendre un logiciel entier pour concevoir une ou deux applications. Pour une automatisation de tâches manipulant un ou même plusieurs fichiers, le recours aux fonctions de fichiers du traitement de texte ou du tableur peut être préférable, si c'est là l'outil quotidien de l'opérateur en question et même si ses fonctions en seront tirées à leurs limites. Par contre et contrairement aux gros systèmes, à la portée des seuls informaticiens à même de les programmer, les logiciels horizontaux pour micro-ordinateurs permettent toujours une informatisation partielle et ils sont à la portée de tous, pour leurs commandes les plus simples. Cette facilité occulte souvent les besoins en programmation par des informaticiens ou des amateurs éclairés en programmation. Un programme sur mesure automatisera complètement la gestion des logements. Même s'il a été bricolé par un amateur, ce programme sera plus efficace que la sous-utilisation des commandes simples d'un gestionnaire de bases de données. Il faut apprendre aux secrétaires quand demander ce type de programme.

Enfin, assimilons cavalièrement l'ergonomie aux métaconnaissances. Le degré de convivialité des outils doit être exposé. Ainsi le professeur d'informatique donnant cours à des pédagogues se fera l'écho de P. Beguin (1991). dira que « dans une situation d'enseignement sur informatique, l'utilisation d'un écran pour deux élèves peut être un optimum pédagogique. Or les écrans ne sont en aucun cas conçus pour répondre à de tels besoins ». Ensuite, la réorganisation de tâches professionnelles autour des outils est un contenu de cours capital car elle conditionne l'acceptation des outils les plus avancés, généralement intégrés. Des éléments de recherche opérationnelle et de cybernétique devraient donc être inclus au cursus. La traduction est par exemple à une période-charnière. Le logiciel intitulé « poste de travail du traducteur » postule la circulation électronique ininterrompue des documents, donc une réorganisation du travail en amont et en aval de la traduction, à l'échelle de l'entreprise ou du ministère. Il présuppose également un éclatement du travail en de nouvelles sous-tâches. En effet, le texte à traduire doit être sur support électronique pour permettre la consultation automatique des glossaires et des anciennes traductions du traducteur. Ce support permet aussi la restitution, dans le texte traduit, de la typographie et de la mise en page du texte originel. En livrant leur travail sur support électronique, les traducteurs permettent à leur tour à leur réviseur de relire et commenter ces traductions dans le mode « corrections provisoires » du traitement de texte. Dans ce mode, les corrections sont notées, comme sur la copie d'un élève; elles peuvent donc être soit confirmées soit infirmées par le traducteur donc connues de lui, ce dernier sautant directement de l'une à l'autre par la pression d'une seule touche. Ces sous-tâches, ces réorganisations, doivent être décrites et exercées au cours.

Cours de logiciels entiers ou de tranches transversales?

Grâce à la convivialité accrue des logiciels horizontaux, l'accès à des fonctions complexes est plus aisé. Il reste cependant hors de portée d'une catégorie d'administratifs. En traitement de texte, c'est le cas des publipostages conditionnels,

les formulaires avec macro-commandes, les tableaux auto-calculants (par exemple des résultats qui se transfèrent en cascade d'un tableau à l'autre). Nous pensons que trois niveaux d'enseignement s'imposent pour les logiciels horizontaux. Ces couches requièrent une formation à des volets différents du logiciel ou des logiciels, lorsque l'apprenant doit en connaître plusieurs. Voici ces profils pour le traitement de texte:

MANIPULATIONS EN TRAITEMENT DE TEXTE	CONNAISSANCES REQUISES	DUREE MINIMALE DE LA FORMATION
Programmation d'une application entière	Programmation Langage de macro-commandes du traitement de texte	15 heures pour le langage de macro-commandes Selon le type d'application, 30 à 60 heures d'algorithmique
Concepteur de documents-types, de feuilles de style, de macro-commandes (sans instructions de contrôle)	Commandes avancées du traitement de texte	8 heures
Utilisateur final des applications, documents-types, feuilles, macros et des commandes simples du TT	Commandes courantes du traitement de texte	8 heures

Sans les nommer, Microsoft a distingué ces couches au fil des versions de ses logiciels, par exemple, pour WORD:

- Affichage soit des champs (variables préprogrammées dans le logiciel ou variables de l'utilisateur de types divers) soit de leur contenu.
- Enregistrement de macros (dont les instructions sont alors invisibles aux utilisateurs), ou écriture de leur texte.
- Générateurs de boîtes de dialogue ou écriture de leurs coordonnées graphiques et de leur texte.
- Paramétrabilité des menus et icônes ou choix des valeurs par défaut, etc.

La qualité de l'automatisation d'un service, d'un indépendant, ne devrait pas être tributaire du manque de motivation ou de compétence du personnel. Or c'est souvent le cas en micro-informatique, sous le prétexte de l'adaptation des outils à l'utilisateur « et non l'inverse ». Le rapport coût-bénéfice de l'enseignement doit surtout être pesé pour les personnes au travail.

Enseigner la programmation pour quelques boucles dans quelques macros n'est pas rentable et peut dégoûter les apprenants de toute la micro-informatique. Et pourquoi ne peut-on pas non plus acheter des lettres-types, des formulaires, des macro-commandes pas trop complexes, avec ou sans programmation, comme on commande des applications clé sur porte? Un marché de services s'offre là aux spécialistes de ces logiciels horizontaux. Ce marché est destiné à des non-

informaticiens, une journée d'informaticien étant plus chère qu'un ou plusieurs logiciels. L'implantation des logiciels crée d'autres métiers encore, par exemple la gestion des hypertextes: « Bush imagine même une (...) sorte d'ingénierie des ponts et chaussées au pays des publications, dont la mission serait d'aménager des réseaux de communication au sein du corpus immense et toujours croissant des sons, des images et des textes enregistrés », fait observer P. Lévy (1990). A leur corps défendant, les informaticiens suppriment des emplois. Ne peuvent-ils signaler ceux qui sont apparus, sans libellé et sans offres d'emplois dans les grands quotidiens, et y préparer les apprenants?

ET la programmation?

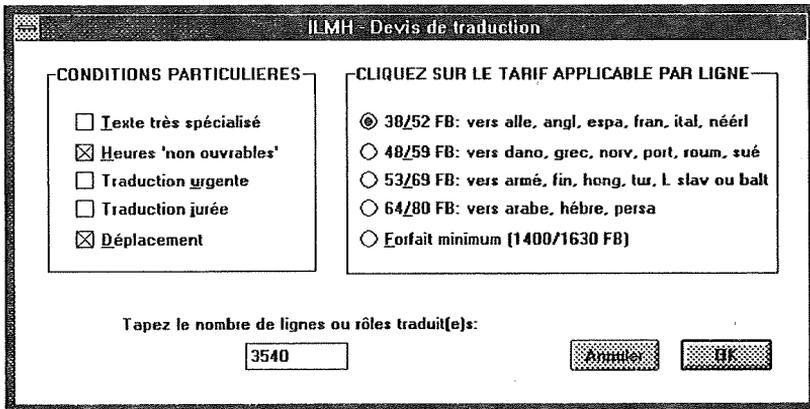
Le niveau des instructions des langages dits de « haut niveau » comme Pascal et Basic était encore assez élémentaire. Cette époque était également celle des pseudo-langages, encore très formels, et des ordinateurs onéreux. Dans le secondaire, l'algorithmique s'enseignait donc sur papier, comme les mathématiques. Le droit aux essais et erreurs, aux débogages auxquels s'adonnent quotidiennement les programmeurs et pour lesquels ils disposent même de routines, était refusé aux débutants en algorithmique et en programmation. Le petit nombre de machines limitait le test des programmes. Aussi, les élèves étaient privés des rétroactions éclairantes mais aussi motivantes que procurent les messages d'erreurs et l'observation des performances des programmes. Il n'est donc pas surprenant que l'enseignement de l'algorithmique et de la programmation dans le secondaire ait été arrêté ou freiné dans plusieurs pays. Pourtant, les changements survenus en micro-informatique invitent les inspecteurs à réintroduire ces cours dans le secondaire ou à les remettre à l'honneur.

De syntaxe moins contraignante que les pseudo-langages, les générateurs d'instructions comme Aladin facilitent tout d'abord un enseignement séparé de la programmation et de l'algorithmique.

Par ailleurs, les instructions qu'inclut le programmeur dans les langages de macro-commandes sont des commandes de logiciels horizontaux, qui sont vraiment de haut niveau. Aussi la programmation d'applications de niveau professionnel devient-elle envisageable, même avec des connaissances limitées en programmation et en algorithmique. Les exercices concevables actuellement peuvent être directement utiles à une profession. A titre d'exemples pour la traduction, les utilitaires listés ci-dessous ont été programmés par un de nos étudiants, programmeur-amateur. Ils sont en WordBasic, pour être disponibles avec cette plume électronique qu'est, pour le traducteur, son traitement de texte:



Voici un autre programme de néophyte, également en WordBasic. Il est utilisé par des traducteurs professionnels car il permet de donner immédiatement un devis par téléphone:



Il contient surtout des multiplications et des demandes de tarifs et de quantités, comme le nombre de pages traduites ou de journées interprétées. Troisième programme de débutant en WordBasic, un convertisseur de mesures anglo-saxonnes. Il est utilisé comme exercice dans notre Institut à un cours d'anglais de première année:

ILMH - Conversion de volumes et de capacités (solides)

CLIQUEZ SUR L'UNITE DEPUIS LAQUELLE CONVERTIR:

Cubic inch
 Gill
 Pint
 Quart
 Gallon
 Peck
 Cubic inch
 Cubic foot

TAPEZ LE CHIFFRE DE LA CAPACITE A CONVERTIR:

Ici encore, il s'agit surtout d'une suite de multiplications. Enfin, voici un générateur de lettres commerciales:

Comment a-t-on connu la société qui fait l'offre de vente?

1. Votre annonce dans la revue spécialisée ... a attiré notre attention sur votre société.
2. Votre offre dans le quotidien/l'hebdomadaire ... du ... nous paraît intéressante/nous a convaincus.
3. A la dernière foire ... votre offre nous a fait une bonne impression.

Présentation de la société

13. A titre d'information, nous vous communiquons quelques données relatives à notre entreprise.
14. Notre société existe depuis ... et opère principalement dans ...
15. Nous fabriquons ...; nos clients sont ...

Motif de la demande

22. Nous voulons élargir notre gamme de produits ... avec des articles bon marché/de prix moyens/de haut de gamme.
23. Jusqu'à présent, nous avons fabriqué des ... en ... (matériau), mais nous aimerions utiliser un autre ... (nom) pour notre production.
24. Nos clients sont très intéressés par ...

Demande d'envoi de documentation

32. Pourriez-vous nous faire parvenir une documentation complète sur ...
33. Nous avons besoin d'urgence, pour nos clients, de ... exemplaires de votre prospectus ... (description exacte).
34. Nous avons besoin de votre prospectus sur ... d'ici le:

Qualité et garantie

54. Avant de vous passer une commande ferme, nous avons besoin de données exactes quant à la qualité.
55. Nous vous prions de nous communiquer des informations quant à la composition du matériau utilisé dans la fabrication.
56. Vous sera-t-il possible, à long terme, de livrer une marchandise de qualité analogue ?

Listes de prix/tarif

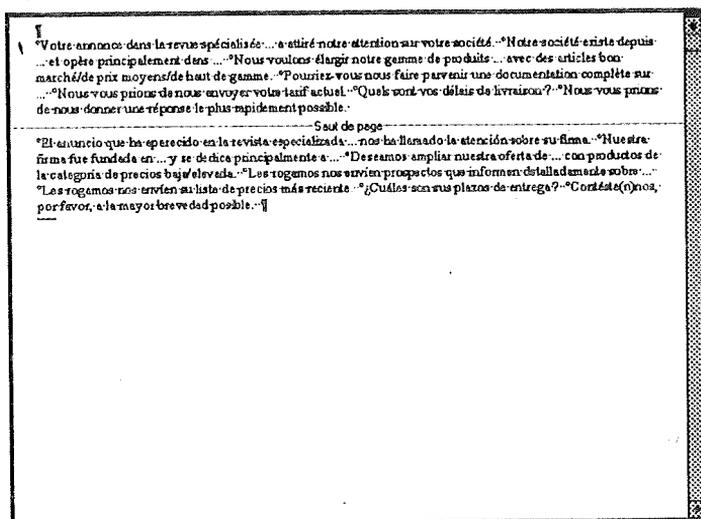
71. Nous vous prions de nous envoyer votre tarif actuel.
72. Quels sont les prix actuels pour ... ?
73. Jusqu'à quand votre tarif est-il valable ?

Emballage et frais d'emballage

91. Nous comptons sur le fait que la marchandise nous parviendra sans dommage. Nous vous prions d'utiliser un emballage protecteur.

TAPEZ LE NUMERO DE LA LIGNE A PRELEVER DE CE TEXTE POUR LA LETTRE PUIS PRENEZ ENTER

Son noyau est constitué de consultations de paragraphes-types (également appelés « insertions automatiques », « paragraphes-types » ou encore « bibliothèques de paragraphes », selon le jargon du traitement de texte). Une fois les fragments désignés par un clic! de l'utilisateur, la lettre écrite par le programme s'affiche dans deux ou plusieurs langues:



nouvelles formations, nouveaux contenus

L'informatique professionnelle se perfectionne sans cesse. L'informatique linguistique force par exemple la linguistique à décrire son objet d'étude avec plus de finesse. C'est ainsi qu'elle a découvert l'existence de 200 conjugaisons des verbes français au lieu des 4 que mentionnaient les grammairiens classiques. Aucune formation universitaire ou post-universitaire n'est pourtant organisée en informatique linguistique, au sein de la francophonie. Celles qui existent ailleurs se comptent sur les doigts de la main.

Les invariants des logiciels sont sémiologiques et un cours de sémiologie serait dès lors utile aux informaticiens eux-mêmes. La grammaire des interfaces graphiques d'utilisation est en cours d'écriture. Certains ouvrages anciens véhiculaient déjà certaines de ses règles, comme le guide de la conception de manuels scolaires de J. Lonergan (1984). L'enjeu de l'idéographie est d'importance, comme l'a vu P. Lévy: « Il ne s'agit pas de faire appel à l'image pour illustrer ou agrémente le texte classique, mais bel et bien d'inaugurer une écriture nouvelle: un instrument de connaissance et de pensée qui soit aussi et intrinsèquement image animée ». On sait notamment que les modèles mentaux sont imagés. L'hypertexte aura lui aussi sa rhétorique. P. A. Carlson (1988) suggère par exemple de concevoir des noeuds de la taille d'un concept.

Beaucoup de facultés universitaires et d'instituts para-universitaires dispensent un cours de philosophie sur les sciences sous-tendant le métier auxquels ils préparent. Ce cours devrait explorer les acquits de l'informatique. Un philosophe du langage filera (c'est-à-dire développera) par exemple la métaphore de l'hypertexte, avec P. Lévy. « dix mille signes ou seulement cinquante repliés derrière un mot ou une icône, des emboîtements compliqués et variables (...) Au rythme régulier de la page succède le mouvement perpétuel de pli ou de dépli d'un

texte kaléidoscope ». Ou encore: « Nelson poursuit le rêve d'un immense réseau accessible en temps réel contenant tous les trésors littéraires et scientifiques du monde (...). *Xanadu*, en tant qu'horizon idéal ou absolu de l'hypertexte, serait une sorte de matérialisation du dialogue incessant et multiple que l'humanité entretient avec elle-même et avec son passé ».

Caroline de Schaetzen

Institut Marie Haps, rue d'Arlon,

111040 Bruxelles- Belgique

Tél.: (02) 511 92 92 Fax: (02) 511 98 37

Bibliographie

- BARON, J.-L. (1990): « L'informatique en éducation, vingt ans après », in *Culture technique*, n° 21, « L'emprise de l'informatique », CRCT, Neuilly.
- BEAU, D. (1983): *100 fiches de pédagogie des adultes à l'usage des formateurs*, éditions d'organisation, coll. « EO/formation permanente », Paris.
- BEGUIN, P. (1991): « L'élève: utilisateur ou consommateur? Quelques réflexions issues d'une approche instrumentale en ergonomie », in *Informatique et apprentissage, Actes de l'université d'été de Châtenay-Malabry, 1990*, Université Paris-Sorbonne et INRP, Paris.
- BESNAIOU, R., MULLER, C. et THOUIN, Ch. (1988): *Concevoir et utiliser un didacticiel*, Les éditions d'organisation, Paris.
- CARLSON, P. A. (1988): « Hypertext: A Way of Incorporating User Feedback », in *Text, Context and Hypertext. Writing with and for the Computer*, The MIT Press, Cambridge and London.
- LELOUCHE, R. (1993): « Essai de synthèse », in *Actes de la troisième rencontre francophone de didactique de l'informatique, Sion, du 6 au 11 juillet 1992*, EPI, Paris.
- LEVY, J.-F. (1995), éd.: *Pour une utilisation raisonnée de l'ordinateur dans l'enseignement secondaire. Analyses de pratiques et propositions pour un meilleur usage des instruments micro-informatiques*, EPI et INRP, Paris.
- LEVY, P. (1990): *Les technologies de l'intelligence. L'avenir de la pensée à l'ère informatique*, La découverte, coll. « Sciences et société », Paris.
- LEVY, P. (1991): *L'idéographie dynamique. Vers une imagination artificielle?*, La découverte, coll. « Textes à l'appui », Paris.
- LONERGAN, J. (1984): *Video in Language Teaching*, Cambridge University Press, coll. « New Directions in Language Teaching », Cambridge, London, New York, New Rochelle, Melbourne, Sydney.
- SFEZ, L. (1990): « Les réalités écraniques à l'école », in *Culture technique*, n° 21, « L'emprise de l'informatique », CRCT, Neuilly.

Quelle informatique pour une université de lettres et sciences humaines? L'expérience de l'université Stendhal. Un bilan dix ans après

Georges Antoniadis & Marie Christine Jandard

présenté le vendredi 12 avril 1996 à 12h00

Résumé: *L'Université STENDHAL est une des quatre universités Grenobloises. Les disciplines principales enseignées sont: les langues, les lettres anciennes et modernes, la communication, les sciences du langage. Les enseignements d'informatique, introduits en 1985, optionnels ou obligatoires, sont dispensés d'une part pour les étudiants de première à la quatrième année, d'autre part, sous forme de stage, pour le personnel enseignant et administratif de l'université ainsi que pour les étudiants préparant une thèse. La grande majorité des étudiants qui rentrent à l'université Stendhal n'ont pas des connaissances informatiques préalables.*

Quatre objectifs ont été fixés dès 1985:

- a) dispenser un enseignement de base (théorie plus pratique) en première et deuxième année,*
- b) mettre en place au niveau de la troisième et quatrième année des enseignements ou des cursus spécifiques d'informatique, en étroite liaison avec la discipline principale des étudiants,*
- c) sensibiliser le personnel enseignant et administratif de l'université aux possibilités mais aussi aux limites de l'informatique,*
- d) créer une « spécificité informatique » pour Stendhal qui s'appuie sur les disciplines de l'université.*

Dix ans après, nous nous proposons d'exposer notre expérience en mettant en avant trois points essentiellement:

- a) l'originalité et les caractéristiques de notre public,*
- b) l'(in)adéquation entre l'enseignement de l'informatique et les attentes de notre public,*
- c) notre démarche pour que l'informatique ... soit à une université littéraire.*

Nous aimerions confronter notre démarche avec l'approche d'autres pédagogues intervenant auprès d'un public similaire et trouver matière dans ces échanges pour une définition de « l'informatique pour non spécialistes ».

1. L'environnement

L'Université STENDHAL est une des quatre universités Grenobloises. Elle compte, environ, 7200¹ étudiants. Les disciplines principales enseignées sont: les

¹ Chiffres pour 1995/96

langues, les lettres anciennes et modernes, la communication, les sciences du langage. Comme dans les autres universités françaises, les études sont organisées en trois cycles: la première et la deuxième année constituent le 1^{er} cycle (DEUG²), la troisième et quatrième le second cycle (Licence, Maîtrise), le 3^e cycle commence en cinquième année avec le DEA³ et se poursuit avec la préparation du Doctorat.

Le public de Stendhal est majoritairement féminin. Un grand nombre de ces étudiants possède un baccalauréat littéraire et vient à Stendhal pour poursuivre des études dans ce domaine. Il s'agit souvent de personnes à qui les matières scientifiques « font peur » car, pour bon nombre d'entre eux, elles ont été cause d'échec dans leur scolarité. Plusieurs préparent un métier d'enseignement, d'autres visent l'industrie et comptent utiliser pour cela leurs connaissances linguistiques ou en communication.

L'informatique a été introduite à Stendhal en 1985 avec la création des deux premiers postes d'enseignants informaticiens (les auteurs). Depuis cette date la totalité des enseignements d'informatique offerts est assurée par le Service d'Informatique Pédagogique (SIP), service autonome, qui intègre l'ensemble des enseignants d'informatique, titulaires et non titulaires, affectés à l'université. Le SIP intervient à plusieurs niveaux et dispense des enseignements d'informatique optionnels ou obligatoires. Il compte actuellement 17 personnes (enseignants informaticiens, ingénieurs, administratifs), permanents ou contractuels.

2. Quelle informatique pour Stendhal?

Dès 1985, un premier constat s'impose: la très grande majorité des étudiants qui rentre à l'université Stendhal n'a aucune connaissance informatique préalable. Il s'agit donc de mettre en place des enseignements à partir du niveau d'initiation, pour un public poursuivant un cursus de non-spécialiste en informatique. En 1996, ce constat reste toujours valable, l'évolution dans ce domaine, contrairement à ce qu'on pouvait attendre, étant pratiquement négligeable.

Quatre hypothèses (postulats) ont constitué les fondements de la mise en place des premiers enseignements d'informatique:

a) Il s'agit d'enseigner l'informatique « science ». En ce sens exposer et faire comprendre les notions fondamentales⁴ de celle-ci, sa problématique et ses outils doit être notre but principal. Chaque enseignement doit faire progresser l'assimilation de ces notions, faire discerner les possibilités et les limites de l'informatique. Des enseignements « utilitaires », comme par exemple des

² D.E.U.G.: Diplôme d'Etudes Universitaires Générales

³ D.E.A.: Diplôme d'Etudes Approfondies

⁴ Comme les notions de « traitement », « forme », « traitement automatique », « traitement de la forme », ...

présentations des logiciels en quelques heures, n'apportent rien aux apprenants sinon l'illusion d'une simili-connaissance.

b) Il faut associer théorie et pratique. Le rôle de la théorie est d'exposer, expliquer, justifier les concepts de l'informatique, le rôle de la pratique de les concrétiser, les faire sentir par des applications, les faire comprendre par la manipulation (l'exemple). En ce sens, enseigner un logiciel consiste à prendre prétexte pour enseigner à travers lui ces notions.

c) L'enseignement de l'informatique doit être pensé globalement et à priori pour chaque filière de l'université et non au coup par coup en fonction de la demande. En d'autres termes notre rôle n'est pas celui d'un prestataire de services mais d'un partenaire partageant les préoccupations pédagogiques de chaque filière. Pour cela il convient d'être source de propositions d'enseignements d'informatique et d'explications de leur apport et de leur portée.

d) Enfin, au même titre que d'autres enseignements, ces enseignements demandent une durée et des conditions matérielles appropriées. Pour notre part, nous avons mis en place des enseignements d'initiation d'une durée minimale de 45 heures et des groupes de travail n'excédant pas 20 personnes, avec, au plus, deux étudiants par poste de travail.

En fonction de ces postulats, valables encore aujourd'hui, quatre objectifs ont été fixés à la création du SIP en 1985:

comprendre l'informatique

Partant du constat d'une culture informatique inexistante, pour nos étudiants, les enseignements en première et deuxième année (DEUG) ont pour but d'apporter des savoirs et des savoir-faire de base. Il s'agit de les initier à l'informatique, leur faire comprendre un certain nombre de notions fondamentales et les amener à être autonomes face à un système micro-informatique composé de logiciels standards. Il importe de former, pendant ces deux années, des utilisateurs de l'informatique, capables d'une utilisation critique et discernée, ce que nous appelons des « utilisateurs avertis ». Le but de cette démarche est double: d'une part offrir un bagage informatique pouvant être exploité à profit, pour ceux qui vont quitter l'Université ou qui ne poursuivront pas ces enseignements par la suite, d'autre part, constituer les bases nécessaires sur lesquelles seront bâtis des enseignements spécifiques en troisième et quatrième année, enseignements associant l'informatique à d'autres disciplines.

Pendant ces deux premières années l'enseignement d'informatique se fait sous forme de cours magistraux (CM) associés à des travaux dirigés (TD)⁵. En première année l'enseignement comporte 1 heure de CM et 2 heures de TD par semaine sur un semestre; en deuxième année le volume horaire double, puisqu'on passe à 1 heure de CM et 2 heures de TD par semaine sur toute l'année.

⁵ Ce qui revient à associer « théorie » (CM) et « pratique » (TD).

En première année l'enseignement comprend: en CM, la présentation des objectifs de l'informatique, de ses domaines d'application, de sa problématique et de quelques méthodes et outils, l'explication de la terminologie de base, la description matérielle et fonctionnelle d'un système informatique; en TD, l'étude d'un système informatique, du système d'exploitation ainsi que l'étude d'un logiciel de traitement de texte; cette étude nous donne l'occasion de concrétiser, par un sujet faisant partie à priori des préoccupations de nos étudiants, la problématique et les méthodes exposés en CM. En deuxième année nous présentons en CM, d'une part, l'importance, la problématique, la méthodologie et les fonctionnalités des bases de données, d'autre part, les notions de base de l'algorithmique. Les TD se composent de l'étude d'un tableur, de la conception de bases de données (étude d'un S.G.B.D.), d'une initiation à la programmation (Pascal ou Prolog). Ce contenu se complète par des notions sur la structure et l'exploitation du système informatique. Ici, aussi, notre but est de faire passer, par l'exemple, la compréhension des notions fondamentales de l'informatique.

Ce contenu d'initiation se justifie par le but recherché. Il s'agit de montrer ce qu'est l'informatique, faire comprendre sa définition: *science du traitement automatique de formes d'information*. L'enseignement de ceci passe par l'explication de deux concepts: « forme » et « traitement automatique ». Enseigner des logiciels de traitement de texte, tableurs ou SGBD permet, pour nous, de faire assimiler le premier concept et de montrer des exemples d'application du second. Le choix de ces logiciels n'est pas dû au hasard; ils sont les plus utilisés, actuellement, dans le monde du travail et, surtout, ils concernent des domaines faisant partie du champ de réflexion de notre public, manipulant des concepts familiers pour lui. L'initiation à l'algorithmique et à la programmation complète la compréhension du second concept; indispensable, à notre avis, elle permet de montrer, de l'intérieur, ce que « traiter automatiquement la forme » veut dire, tout en faisant prendre conscience de la complexité d'élaboration des productions informatiques.

Ainsi ces deux premières années, par la connaissance des possibilités et des limites de l'informatique, jettent les bases de son appropriation par les autres disciplines de l'université Stendhal.

intégrer l'informatique

S'il est actuellement admis que l'informatique peut être « utilisée » à profit par la plupart des disciplines, pour un grand nombre d'applications, il reste néanmoins vrai que sa simple juxtaposition est insuffisante si on veut que son apport soit significatif et durable. Utiliser l'informatique par une autre discipline ne portera ses fruits que si on parvient à prendre en compte la problématique de la discipline hôte, la faire élargir et donner naissance, ainsi, à des produits innovants. Une telle démarche suppose la conjugaison de deux parties avec comme but l'élaboration d'une problématique nouvelle, différente d'une simple addition de deux problématiques d'origine. En ce sens la connaissance, voire la maîtrise, de chacune des deux disciplines est indispensable. C'est de cette façon, à notre avis,

qu'on pourra mettre au jour de nouveaux domaines, nécessaires à la résolution des problèmes pour lesquels aucune des deux disciplines d'origine ne peut apporter de réponse.

Notre deuxième objectif concerne l'intégration de l'informatique (science) aux disciplines principales de notre Université: Langues, Lettres, Communication, Sciences du Langage. Le but et les modalités de cette intégration sont ceux définis plus haut. Elle suppose des connaissances en informatique et dans la discipline hôte aussi bien de la part des étudiants que des enseignants formateurs. En général elle n'est possible que si ces derniers pratiquent cette intégration déjà pour leurs activités de recherche. En fonction de ces remarques nous avons mis en place des spécialisations incorporant l'informatique, pour quelques filières, à partir de la troisième année (Licence). Cette spécialisation se poursuit en quatrième année (Maîtrise) et, partiellement, au delà.

Une telle intégration n'est pas une tâche évidente. Elle demande une volonté et des efforts, pas uniquement de la part des enseignants informaticiens mais aussi de la part des spécialistes des disciplines partenaires. C'est un processus lent qui suppose que les deux parties, non seulement arrivent à fixer des objectifs communs mais aussi à se forger un langage de communication et de travail. C'est ce dernier point qui semble le plus difficile à réaliser. Rien en effet ne prédispose à cela. La différence de formation et de culture des partenaires rend la tâche délicate. Seule la conviction d'un gain (enrichissement) mutuel peut rendre ce processus viable.

sensibiliser à l'informatique

Vouloir développer des activités d'enseignement d'informatique au sein d'une université littéraire ne peut se concevoir que si le personnel de l'établissement - enseignants, administratifs, personnel technique - coopère à cette tâche, ou au moins n'est pas hostile, car convaincu de l'apport, de la nécessité et de la portée d'une telle démarche. Ceci ne peut être possible que si l'informatique fait partie de leur environnement, de leur réflexion, de leur travail. S'agissant de l'université Stendhal la mise en place des premiers enseignements d'informatique est décidée à une époque où l'outil informatique est pratiquement absent en son sein. Il est donc important que l'enseignement de l'informatique ne s'adresse pas seulement aux étudiants mais accompagne aussi l'informatisation de l'établissement en formant le personnel avec les mêmes buts que ceux recherchés pour les étudiants. Dans cette optique, notre troisième objectif consiste à mettre en place des formations pour le personnel, sous forme de stages d'initiation et/ou de perfectionnement et un suivi des besoins, dans le but d'enseigner pour les uns (administratifs, techniciens) leur outil de travail, faire comprendre pour les autres (enseignants) les conditions et l'apport de l'intégration de l'informatique à leur discipline.

Même si le contenu de ces formations est similaire à celui des étudiants, l'approche pédagogique est sensiblement différente. Elle tient compte des

spécificités et des attentes particulières d'un tel public: plus attentif et réceptif, demandeur, susceptible, avec des idées préconçues, cherchant une rentabilité immédiate, etc. Réussir ces enseignements, mis à part la satisfaction professionnelle et l'accomplissement d'une de nos missions, est une assurance et parfois un préalable pour que l'informatique puisse être acceptée, être comprise, se développer et, enfin, s'intégrer aux autres disciplines.

créer une « spécificité informatique »

Il y a un paradoxe à enseigner l'informatique à des non-spécialistes à l'Université. En général les enseignements qu'on y dispense le sont dans un but de formation de spécialistes d'une science ou d'un domaine; il s'agit d'inculquer les derniers développements de la discipline par des enseignants spécialisés. Dans le cas de l'informatique pour les non-spécialistes, de surcroît littéraires, l'enseignement consiste à présenter les fondements de cette science à un public qui considère cette matière, au mieux comme un complément ou un atout dans la recherche d'un emploi. En d'autres termes, il s'agit d'enseigner les bases d'une science à des personnes, à priori, non intéressées par la problématique de celle-ci mais par ses « aspects utilitaires ». Une telle situation ne peut aboutir à terme qu'à un désintéressement de la part des enseignants spécialistes qui, le plus souvent, chercheront d'autres lieux, plus favorables, pour mener une réflexion sur leur science. Il s'agit en fait d'un aspect de la « vérité », souvent énoncée: *il n'y a pas d'enseignement supérieur sans recherche associée.*

Dans un environnement « Lettres et Sciences Humaines » comme celui de Stendhal, et dans la mesure où on suppose - pour diverses raisons - que la formation de spécialistes en informatique n'a pas lieu d'être, une solution s'impose alors, l'intégration de l'informatique aux disciplines de l'établissement pour la création de nouvelles spécialités. Ces spécialités sont, d'ailleurs, le plus souvent impossibles à proposer dans les établissements scientifiques par manque de préoccupations pour les sciences, dites, non-exactes. Cette démarche d'intégration va dans le sens de l'évolution des différents domaines (et des produits attendus); de plus en plus la composante « Sciences Humaines » est considérée comme indispensable et naturelle pour les produits des sciences exactes. Agir de la sorte revient donc à créer une spécificité informatique pour l'établissement qui s'appuie sur ses forces vives.

Notre quatrième objectif vise la création d'une telle spécificité. Il suppose l'existence de pôles de recherche pluridisciplinaire et une réflexion sur l'émergence des nouvelles problématiques. Compte tenu de l'environnement local, trois domaines font partie de cette spécificité: le traitement automatique de la langue, les nouvelles technologies éducatives pour l'enseignement des langues et les produits de la communication informatisée. Ces trois domaines sont fortement connexes. Ils prennent appui sur une réflexion globale qui concerne l'élaboration des produits informatiques manipulant le langage dans ces différentes formes, ce qu'on appelle les « Industries de la Langue ».

3. Quel bilan dix ans après?

Il ne s'agit pas ici de porter un jugement sur la « justesse » des objectifs fixés, ni sur la réussite ou non des buts assignés. Nous voulons présenter un bilan quantitatif et qualitatif, base pour une discussion sur la difficulté/facilité d'atteindre ces objectifs et les problèmes rencontrés lors de la mise en oeuvre de notre démarche pour les atteindre.

la situation actuelle

Dix ans après avoir fixé les objectifs ci-dessus l'enseignement de l'informatique à Stendhal est présent, sous différentes formes, aux trois cycles d'études ainsi qu'en formation continue. 1200 étudiants, environ, poursuivent un tel enseignement sur les 7200 que compte l'université (données 1995/96).

Au premier cycle (1^e et 2^e année) il constitue une matière obligatoire pour un certain nombre de filières, une matière optionnelle pour les autres. Dans ce dernier cas, le nombre d'étudiants inscrits est d'autant plus important qu'il existe une possibilité de poursuite au niveau du second cycle. La demande est croissante, surtout en première année, sans que celle-ci puisse être complètement satisfaite, faute de moyens. Les notions acquises pendant ces deux années sont souvent utilisées au cours de leur scolarité (confection de rapports, recherche d'informations, exploration de textes, ...). Certains étudiants « découvrent » la nécessité de faire de l'informatique tardivement dans leur cursus (troisième ou quatrième année); des cours d'initiation spécifiques leurs sont alors proposés.

Au deuxième cycle (troisième et quatrième année) l'informatique est présente dans quatre filières: Langues Etrangères Appliquées (LEA), Communication, Sciences du Langage et Langues. Il s'agit soit d'enseignements spécifiques intégrés dans la filière (gestion et recherche d'informations, traduction assistée par ordinateur, pour la filière LEA), soit d'une spécialisation possible: « Communication Informatisée » pour les étudiants de Communication, « Industries de la Langue » pour ceux en Sciences du Langage ou Langues. Ces spécialisations sont le fruit et la concrétisation de l'intégration de l'informatique aux autres sciences de Stendhal. Elles ont été possibles grâce à la volonté et aux compétences pluridisciplinaires des enseignants des secteurs concernés.

En troisième cycle (cinquième année et au-delà) il est possible de suivre des enseignements et préparer un Doctorat dans deux domaines en rapport avec l'informatique. Il s'agit du « traitement automatique de la langue écrite et de la parole » et de la « communication homme - machine ». Ces diplômes résultent de la collaboration entre différentes composantes de l'université. Ils ont été possibles grâce à l'existence d'équipes de recherche dans ces domaines. Signalons, enfin, que des stages d'initiation à l'informatique sont proposés périodiquement aux étudiants inscrits dans les autres DEA et Doctorats de l'université.

Le personnel enseignant, technicien et administratif de l'université bénéficie, chaque année, de stages d'initiation et/ou perfectionnement en rapport avec les tâches qu'il assume ou des projets qu'il compte mettre en oeuvre. Depuis quelques années nombre de ces stages est assuré par des intervenants extérieurs.

former sans déformer

Dix années d'enseignement à Stendhal nous ont permis d'étudier les caractéristiques de notre public littéraire, ses réactions face à l'informatique ainsi que les attentes et les espoirs que cette nouvelle matière suscite.

Un premier constat s'impose: l'informatique ne laisse pas indifférents les étudiants de Stendhal. Ils peuvent l'aimer ou la détester, ils peuvent la trouver compréhensible ou ésotérique, ils peuvent avoir des idées précises sur la question ou une vague conception, néanmoins tous s'accordent pour dire que l'informatique est « utile », qu'il vaut mieux avoir suivi quelques cours, qu'il est préférable qu'elle fasse partie du C.V. Cette approche « pragmatique » n'est pas sans conséquences pour son enseignement: tout autre cours, pensent-ils, faisant une quelconque référence à l'informatique est bon à suivre et équivalent au « cours officiel »⁶. Comme il n'y a pas d'a priori, tout enseignement d'informatique sera bien accepté. Ceci, joint au fait que peu d'étudiants ont des connaissances informatiques préalables, implique qu'il y a peu d'idées fausses à rectifier, pas de mauvaises méthodes de travail à corriger, une bonne acceptation des exercices amenant progressivement à la compréhension de chaque concept. Peu de souhaits de formation à un champ précis de l'informatique sont exprimés; la demande de formation à la programmation, qui au départ était fréquente, a complètement disparu au fil des années.

Si parfois l'informatique est rejetée car assimilée aux mathématiques, le plus souvent elle joue un rôle de catalyseur pour celles-ci; revoir et comprendre quelques notions de mathématiques procure une grande satisfaction, permet de prendre la revanche sur cette discipline, montre que la compréhension d'un concept est facilitée lorsqu'on arrive à montrer son utilité et son utilisation. Cette situation par rapport aux mathématiques, presque anecdotique, ne doit pas éluder l'essentiel des difficultés de perception: l'abstraction comme l'appropriation et l'utilisation du raisonnement analytique. Toute situation qui demande une décomposition systématique est source de difficultés; écrire une macro ou un script, prévoir l'état d'un système informatique après quelques commandes, etc. se fera, au début, dans la douleur. Il est vrai que les systèmes comportent peu de références aux concepts familiers des étudiants en lettres et sciences humaines. Du coup la symbiose avec le système devient inconcevable. Passer de « utilisateur esclave » de la machine à « utilisateur partenaire » puis à « utilisateur maître » le processus est long. Sa place et son rôle face à elle sont difficiles à trouver: *est-ce moi qui fait?, la machine sans*

⁶ C'est ainsi qu'il a fallu pendant des années expliquer que le cours d'informatique était différent du cours de dactylographie utilisant des micro-ordinateurs.

moi?, moi et la machine? etc. La juxtaposition du verbe *faire* à lui-même n'est pas une opération naturelle pour eux. Peut-être parce que la machine n'est pas située à sa juste valeur; trop humanisée, elle est sensée être capable de saisir l'implicite. L'affirmation très souvent énoncée, *la machine n'est pas intelligente*, ne fait qu'accentuer cette croyance puisqu'elle l'humanise davantage. On s'attend donc à ce qu'elle possède les mêmes connaissances que soi, au moins en partie; qu'elle interprète, de la même façon, l'implicite. Evidemment l'expérience n'est pas concluante et la méfiance, due à l'incompréhension, s'installe.

La méfiance et les difficultés d'abstraction conduisent les étudiants à ne croire et comprendre que le visible, le palpable. Toute action qui produit un résultat tangible est, généralement, comprise; par contre celles sans effet immédiat ne seront pas considérées. Le résultat d'une telle démarche est aisément prévisible; des actions consécutives, à effet visible, produiront des résultats déconcertants pour l'utilisateur car non attendus par lui. Le désarroi est d'autant plus grand que la cause de l'inattendu est lointaine dans le passé; ainsi, par exemple, la manipulation des répertoires et des fichiers, le maniement des index, etc., si ils ne sont pas clairement compris ou visualisés, peuvent conduire à une interprétation erronée des réactions du système aux sollicitations futures de l'utilisateur.

La conséquence d'une telle situation est, apparemment, paradoxale pour des spécialistes de la langue. Les difficultés de compréhension se matérialisent au niveau de leur expression et de l'interprétation des messages fournis par la machine (d'ailleurs souvent obscurs et ambigus, n'apportant aucun élément de réponse). Parler d'informatique devient laborieux, même à un niveau technique moindre; son vocabulaire mal maîtrisé rend l'expression gauche et les énoncés presque incompréhensibles. Le discours manque manifestement de cohérence vu que les mots employés n'ont pas de référents bien déterminés, ne se rattachent pas aux concepts courants du locuteur. Ainsi démuné d'outils pour comprendre, de repères pour se situer et sans possibilité de s'exprimer il est normal que l'étudiant en lettres se sente étranger à l'informatique. Son enthousiasme initial tombera vite laissant la place à l'impression d'être incapable de « se sentir informaticien ». A partir de là toute intégration de l'informatique devient illusoire. Des nouveaux métiers, comme par exemple ceux liés aux industries de la langue, ne peuvent pas avoir une réalité car même leur expression est impossible.

L'inadéquation constatée, pour notre public, d'un enseignement selon le processus habituel: compréhension des concepts, appropriation de la machine, appropriation du discours informatique, nous a amené à reconsidérer ce processus et à en forger un nouveau, adapté aux caractéristiques et aux préoccupations de nos étudiants. Il s'agit de s'appuyer sur leur connaissance première, le langage, pour introduire et consolider les concepts de l'informatique. Ainsi le processus d'apprentissage mis en place débute par l'appropriation du langage (discours) informatique. Le but est de leur faire sentir qu'ils peuvent utiliser des concepts connus auxquels les nouveaux termes peuvent être rattachés. L'expression

informatique s'acquiert ainsi par référence à un monde de connaissances qui leur est familier et qui leur permet de mieux appréhender ces nouvelles notions. La maîtrise de l'expression informatique amène petit à petit la confiance, fait situer la machine à sa juste valeur d'exécutant, l'assimile au distributeur de boissons qu'ils utilisent couramment. Du coup, la comprendre devient plus familier et plus abordable, accessible avec leurs concepts. Le contact est établi, la symbiose est maintenant possible. La « machine exécutant » fait ouvrir à d'autres formes de réflexion, à d'autres problématiques considérées et vécues comme le prolongement de la leur. Ce ne sont pas des choses ou des méthodes nouvelles qu'il faut apprendre ce sont des notions qui découlent, ou presque, de celles déjà assimilées, qui font référence à un monde de connaissances connu et qui l'enrichissent, le complexifient et le complètent. L'intégration de l'informatique, alors, commence à devenir possible.

4. Conclusion

Aujourd'hui, dix ans après, l'informatique fait partie, presque naturellement, du paysage de Stendhal. Elle concerne un grand nombre d'étudiants et elle est jugée « nécessaire » par bon nombre de collègues non informaticiens qui en sont maintenant demandeurs. Elle arrive à avoir un statut et une reconnaissance ne serait-ce que physique par des locaux appropriés. Elle a réussi à se faire comprendre, en partie, dans la mesure où elle n'est plus assimilée à une dactylographie fonctionnelle. Sa problématique commence à être perçue même si encore ses aspects utilitaires priment. Petit à petit elle n'est plus considérée comme « le vernis qu'on doit avoir » mais comme un autre regard possible pour des problèmes que l'on se pose. Son appropriation aussi bien par les étudiants que les autres enseignants a commencé, même si elle est encore timide et chancelante.

Cet optimisme mesuré ne doit pas dissimuler le chemin qui reste à parcourir. L'informatique prendra pleinement son essor à Stendhal que lorsqu'elle sera reconnue (comprise) en tant que science au même titre que les autres disciplines. Elle a des atouts pour cela (ses aspects applicatifs) mais aussi des exigences (ne serait-ce que matérielles). Seule son intégration par les autres disciplines permettra de tirer pleinement profit de ses possibilités et, surtout, permettra aux étudiants de Stendhal d'être convaincus qu'ils ont quelque chose à dire et à apporter dans ce domaine.

Georges Antoniadis & Marie Christine Jandard
Service d'Informatique Pédagogique, Université Stendhal
B.P. 25 38 040 Grenoble Cedex 9, France

Une première expérience de brain-storming en T.D. d'informatique

Patrice de MARCONNAY

Présentée le vendredi 12 avril à 11h00

Résumé: Cet article présente quelques expériences d'enseignement effectuées par un enseignant presque débutant en T.D. d'informatique auprès d'étudiants en premier cycle universitaire. Elles visent à évaluer le bénéfice d'une légère amélioration de la relation enseignant - étudiant et à favoriser le processus cognitif d'apprentissage par une technique adaptée de brain-storming. Bien que les résultats soient incomplets et subjectifs, ils semblent indiquer que ces techniques (en particulier le brain-storming) sont utiles dans le contexte d'étudiants peu motivés.

1 Introduction.

L'enseignement d'une matière obligatoire en premier cycle universitaire n'est jamais aisé. Dans le cas d'un T.D. d'informatique, on peut penser que la matière est suffisamment attractive et "dans le vent" pour être bien perçue par tous les étudiants. Il n'en est rien: comme toutes les autres disciplines, l'informatique a besoin de méthodes et de trucs qui vont permettre à l'enseignant de mieux communiquer son savoir, mieux motiver les étudiants pour un apprentissage plus efficace et une compréhension plus profonde. Dans ce cadre, nous avons essayé deux méthodes pour améliorer l'apprentissage:

1. La « friandise », qui consiste ni plus ni moins à apporter des friandises aux étudiants pour renforcer la relation enseignant-étudiant dans le contexte de l'analyse transactionnelle;
2. Le « remue-méninges », une forme de brain-storming utilisée pour favoriser l'apprentissage.

Ces méthodes les ont été expérimentées par un moniteur de l'enseignement supérieur, dont les activités pédagogiques antérieures se limitaient à neuf mois d'enseignement dans des organismes publics de formation continue pour adultes.

La première partie de ce papier présente le contexte d'enseignement dans lequel ce travail a été effectué: au sein de l'université Stendhal - Grenoble III, au Service d'Informatique Pédagogique par un moniteur de l'enseignement supérieur. Dans la partie suivante, les expériences d'enseignement (la friandise et le remue-méninges) sont décrites en détail avec les concepts sur lesquelles elles reposent, les objectifs et les modalités d'application. S'ensuit une évaluation subjective et objective de ces deux expériences. Enfin, nous concluons sur les possibilités de ces deux techniques et ouvrons quelques perspectives.

2 Contexte d'enseignement.

2.1 L'université Stendhal - Grenoble III.

L'université Stendhal - Grenoble III est, en effectif, la plus modeste des trois Universités de Grenoble (France). Elle accueille des étudiants dans les disciplines de Lettres, Langues Vivantes, Linguistique et Communication.

L'Université Stendhal est regroupée sur le campus universitaire de Grenoble avec une petite délocalisation à Valence. Elle comprend plus de 7000 étudiants (dont 420 à Valence, et 700 étudiants étrangers), 315 enseignants - chercheurs et 150 personnels administratifs et techniques. Elle est composée de cinq Unités de Formation et de recherche (U . F . R .)

Depuis quelques années, des enseignements d'informatique ont fait leur apparition dans les formations, d'abord SOIIS forme d'option facultative, puis sous forme de cours obligatoire. Pour répondre à cette demande d'enseignement en informatique, un service commun aux différentes U.F.R. a été créé: le Service d'Informatique Pédagogique .

2.2 Le Service d'Informatique Pédagogique.

Le Service d'Informatique Pédagogique (S.I.P.) a pour mission d'assurer des formations en informatique pour les étudiants et le personnel des différentes U.F.R. de l'université. Ceci inclut la mise en place et la gestion de quatre salles informatiques, ainsi que la charge d'enseignement associée aux cours dispensés, à la fois sur les sites de Grenoble et Valence.

En 1994-1995, 2414 heures d'enseignement ont été dispensées auprès de 978 étudiants (environ 10% des étudiants de l'université) dans des filières de:

- premier cycle: Deug 1ère et 2ème année et IUP 1ère année;
- second cycle: licence et maîtrise LEA, licence EAO, IUP Communication, Traitement Automatique de la langue;
- troisième cycle: école doctorale.
- formation continue: personnel de l'Université;
- plan Informatique Pour Tous (I.P.T.).

Les ressources humaines du SIP sont composées de 4 maîtres de conférences, 1 P.R.C.E., 3 moniteurs-allocataires, 11 vacataires, 2 personnels administratifs et techniques, 2 contractuels, et 1 statutaire, soit un total de 24 personnes.

C'est dans le cadre du Monitorat que j'ai été affecté depuis septembre 1993 au SIP.

2.3 Le monitorat d'Enseignement Supérieur.

Nous allons maintenant présenter rapidement le Monitorat.

2.3.1 Cadre et objectifs.

Le Monitorat d'enseignement supérieur a été mis en place en 1989. Il est destiné à apporter à des allocataires de recherche une initiation à l'Enseignement Supérieur par une expérience d'enseignement et un programme de formation. Le moniteur recruté par un des Centres d'Initiation à l'Enseignement Supérieur (C.I.E.S.) s'engage à assurer un service d'enseignement de 64 h TD par an (1/3 de service de maître de conférences) sous la direction d'un tuteur pédagogique. Il doit aussi participer aux stages d'initiation organisés par le C.I.E.S. - environ dix jours par an.

2.3.2 Formation.

Les formations mises en place par le C.I.E.S. de Grenoble sont résumées dans le tableau ci-dessous.

N°	Thème des principales formations	nbre de jours
-	Introductions au métier d'enseignant	1
-	Introductions au système universitaire et à ces enjeux	4
F3	Techniques de communication (niveau I)	3
F4	Pédagogie des disciplines	2
F5	Techniques de communication (niveau II)	3
-	Didactique et pratiques pédagogiques	2
F7	Pratiques pédagogiques en informatique	2
F8	Technique de communication (niveau III)	3
Total de ces formations: 20 jours.		

Les formations traitant de la communication (en particulier F3 F5 et F8 décrites plus précisément en annexe A.1) étaient particulièrement intéressantes. Ce point sera précisé dans la suite.

2.4 Enseignements dispensés

Les enseignements qui nous intéressent ont eu lieu entre 1993 et 1995. Il s'agit de trois T.D. d'informatique dont les principales caractéristiques sont données ci-dessous:

N°	Période	Profil des étudiants	Volume	Responsable	Type d'enseignement
M2	2 ^e semestre 93-94	L.E.A. 1 ^{ère} année	56 h	Mme M. H. Stéfanini	T.D.
M3	1 ^{er} semestre 94-95	S.L. 1 ^{ère} année	24 h	Mr. G. Antoniadis	T.D.
M4	1 ^{er} semestre 94-95	I.P.T. (divers profils)	24 h	Mme V. Guéreau	T.D.

Volume horaire total: 104 heures

Une présentation plus détaillée des conditions de chacun de ces enseignements se trouve en annexe A.2.

Au S.I.P., les enseignants jouissent d'une grande liberté de présentation et d'organisation. Cette liberté est complétée par le fait que chaque enseignant dispensant un T.D. organise et corrige les examens de ses étudiants. Seuls les contenus des T.D. sont soigneusement établis au cours de réunions pédagogiques préalables. Cette liberté d'action a rendu possible une expérience d'enseignement encore peu répandue à l'Université.

3 Une première expérience d'enseignement.

Le titre de cette section est à double sens. Par "une première expérience d'enseignement", il faut comprendre le mot expérience dans le sens "expérimentation", c'est à dire mise en place de conditions expérimentales dans un but précis, collecte des données et évaluation. Cette expérimentation doit être suivie de quelques autres, permettant ainsi d'infirmer ou de confirmer les résultats de la première expérience. Le deuxième sens de ce titre est de remettre ces modestes expérimentations dans leur cadre: celui d'un enseignant débutant, pour qui ces enseignements constituaient une première expérience. C'est avec ses deux sens qu'il convient de comprendre le titre.

Au cours de la formation reçue dans le cadre du Monitorat se trouvent trois stages traitant plus particulièrement de la communication. Il s'agit des stages F3, F5 et F8 décrits en annexe A.1. Ces stages mettent en avant deux idées fortes:

1. enseigner, c'est communiquer un savoir au travers d'une relation enseignant - étudiant;
2. l'acquisition du savoir passe par un processus d'apprentissage pour lequel il existe plusieurs modélisations.

C'est sur cette base qu'ont été conçues les expériences.

3.1 La relation enseignant - étudiant.

3.1.1 Concepts sous-jacents.

Lors du premier stage de communication (F3), certains concepts de l'analyse transactionnelle ont été présentés. Dans cette modélisation des relations humaines, le rapport Enseignant-Elève est principalement une relation **Adulte-Adulte** et une relation **Parent-Enfant**. La relation **Parent-Enfant** peut être affinée en **Parent Censeur-Enfant** et **Parent Nourricier-Enfant**. Cette décomposition rappelle beaucoup d'impressions vécues tout au long de la scolarité. Pour savoir dans quelle mesure elle est juste, il convient de renforcer son aspect **Parent Nourricier** vis à vis des étudiants, pour voir s'il en résulte un changement sensible dans les relations enseignant - étudiant.

3.1.2 Une méthode simple de renforcement de cette relation.

Le mot Nourricier a été compris au sens propre. L'idée est simple: mettre en place une situation naturelle où l'enseignant apporte à l'étudiant une friandise.

Cette friandise n'est pas la récompense d'une quelconque réussite; elle est proposée à tous, gratuitement. La difficulté réside dans la manière de présenter cette initiative inhabituelle. Il faut que cela paraisse naturel. Et cela nécessite quelques efforts d'imagination de la part de l'enseignant !

Dans notre cas, il s'agit d'un gâteau fait maison par l'enseignant. Profitant d'un T.D. où l'objectif est de mettre en forme sur traitement de texte une recette de cuisine, nous mangeons ensemble le gâteau dont la préparation est décrite dans la recette. Le déroulement du T.D. n'est pas outre mesure perturbé, et il n'y a pas de temps perdu .

Essayer de renforcer la relation enseignant - étudiant n'est pas tout: favoriser le processus d'apprentissage est encore plus important.

Le remue-méninges. Seuls les groupes M3 et M4 ont participé à cette expérience. Il semble que les objectifs décrits ci-dessus aient été atteints dans les deux groupes. Dans le groupe de M3, l'effet du remue-méninges a été largement positif. La moitié des étudiants ont pris visiblement plaisir à participer. Une véritable dynamique de groupe s'est mise en place entraînant une entraide et une participation plus actives.

D'autres effets bénéfiques ont été constatés, comme par exemple:

- peu d'étudiants en retard (le remue-méninges intervenait en début de T.D.);
- un éveil plus rapide de l'attention en début de cours permettant de démarrer vite le T.D. après le remue-méninges;
- un moment apprécié par les étudiants ayant manqué la séance précédente, qui trouvaient là un moyen de savoir ce qui avait été fait pendant leur absence !
- une forte diminution des questions concernant des notions abordées dans les séances précédentes (au lieu de demander, ils regardaient leurs notes).

Enfin, les notes prises pendant les remue-méninges ont permis de repérer les principales difficultés du T.D., d'évaluer la présentation des connaissances et de faire les adaptations nécessaires.

3.3.3 Evaluation plus objective.

Conscient de la fragilité de ces indices favorables, il fallait chercher d'autres sources plus objectives d'évaluation. Elles concernent l'assiduité en cours (retards et absences), la quantité de travail effectué (pendant et en dehors du T.D.), ainsi que le niveau des connaissances.

- Les retards: très peu d'élèves en retard (quelques retards de moins de 5 minutes, 2 retards de plus de 5 minutes - dont un annoncé à l'avance).
- Les absences: 0,69 absences par étudiant en moyenne dans le groupe M3, 1 dans le groupe M4 (pas encore d'information concernant les autres groupes de T.D.).
- La fréquentation du libre-service: en dépouillant soigneusement le registre d'inscription aux salles libre-service, j'ai étudié la fréquentation des étudiants de S.L. et I.P.T. tous groupes de T.D. confondus. La moyenne pour le groupe M3 est

de 1,75 fréquentations par étudiant, de 4,35 en M4, contre respectivement 0,55 et 1,41 pour les autres étudiants de la même filière - soit un facteur d'au moins 3 ! -. Notons que certains étudiants avaient la possibilité de travailler sur ordinateur ailleurs qu'en libre-service.

· Les notes de T.D.: Elles ne sont pas un bon indicateur car j'ai assuré la création, la surveillance et la correction de l'examen.

· Les notes de C.M.: je n'intervenais pas directement dans ce cours, mais les notes pourraient révéler un intérêt plus prononcé pour l'informatique. Il n'en est rien: les notes sont comparables à la moyenne des autres étudiants.

Ces résultats numériques sont résumés dans le tableau qui suit:

groupes d'étudiants	absences par étudiant	libre-service par étudiant	moyenne des notes de T.D. (sur 20)	moyenne des notes de C.M. (sur 20)	moyenne générale (sur 20)
groupe de T.D. S.L./	0,7	1,8	17,3	8,5	14,3
autres groupes S.L.	?	0,6	16,1	8,6	13,6
groupe de T.D. I.P.T.	1,0	4,4	16,1	11,2	14,2
autres groupes I.P.T.	?	1,4	15,4	11,2	14,0

3.3.4 Autres éléments d'évaluation.

Il existe d'autres points pour lesquels il n'y a pas de résultats quantitatifs. Par exemple si un exercice donné pour la séance suivante était fait ou non. Aucune contrainte particulière n'obligeait les étudiants à les faire. En moyenne ils étaient régulièrement effectués (d'où une forte fréquentation du libre-service). Ou encore la vitesse d'exécution des exercices pendant le T.D., plus élevée dans le groupe M3 que dans le groupe M4. Ou enfin la réalisation d'une fiche technique sur un petit accessoire de Windows (Presse-papier, horloge, ...) qu'ils devaient concevoir et réaliser avant de la distribuer aux autres étudiants. La moitié des étudiants ont rendu une fiche technique, dont certaines de très bonne qualité (deux d'entre-elles sont données à titre d'exemple en annexe A.3). Certains ont éprouvé une certaine fierté lorsque leur fiche a été distribuée aux autres.

3.2 Le processus d'apprentissage.

3.2.1 Concepts sous-jacents.

Il existe quelques modélisations du processus d'apprentissage. L'apprentissage peut être considéré comme un processus à trois étapes fondamentales¹ :

1. une étape de perception: l'information présentée parvient au cerveau;
2. une étape d'évocation, où l'information est représentée de façon personnelle;
3. une étape de restitution qui se manifeste lors de la réutilisation de l'information.

¹ tiré de L. Balico et al. "Approche de la pratique pédagogique de la gestion mentale", rapport de 3ème année, CIES de Grenoble, 1992 .

Nous voulons favoriser l'évocation et la restitution, sans oublier de présenter les informations de la manière la plus claire possible.

3.2.2 Favoriser le processus d'apprentissage: le remue-méninges.

Dans cette optique, une technique de brain-storming a été adaptée. Elle sera dénommée par le terme "remue-méninges". Pendant cet exercice, les étudiants doivent énoncer de mémoire tout ce qu'ils se rappellent de la précédente séance. Les règles sont simples mais strictes:

1. il faut dire tout ce qui passe par la tête, pourvu que cela ait un rapport avec la dernière séance de T.D.;
2. on a le droit de copier, plagier, répéter ce que les autres ont dit;
3. la quantité d'idées compte plus que la qualité;
4. il est interdit de critiquer;
5. l'enseignant n'intervient pas.

Cet exercice dure dix minutes environ. Il a été testé à chaque début de T.D. pour les groupes M3 et M4. Pendant les dix premières minutes, je les observais en prenant des notes; un étudiant volontaire notait au tableau tout ce qui était dit par ces camarades. L'objectif était triple:

- faciliter l'assimilation du T.D.;
- vérifier la compréhension du dernier T.D., afin de pouvoir faire les réajustements adéquats (précision, nouvelle explication, rectification);
- favoriser une dynamique de groupe par une communication orale.

Ces expériences ont relativement bien fonctionné comme semble le montrer les résultats exposés maintenant.

3.3 Résultats et évaluation.

3.3.1 Remarques et précautions.

Je vais essayer de donner des éléments d'évaluation, dans un premier temps subjectifs, puis d'autres peut-être plus objectifs. Comme il est difficile d'être juge et parti, je n'accorde qu'une valeur relative et prudente à mes impressions. Quant aux données quantitatives, elles portent sur un trop faible échantillon pour avoir une valeur statistique. Cependant, ces évaluations sont considérées comme des indices, à défaut de mieux.

3.3.2 Evaluation subjective.

La friandise. Les groupes M2, M3 et M4 ont été testés. Après un délai de flottement pendant lequel les étudiants n'osaient pas prendre du gâteau, l'ambiance s'est un peu décrispée. Le cours n'en a pas souffert. Il a semblé pour les trois groupes que la relation enseignant - étudiant a été renforcée. L'effet a été particulièrement sensible dans le groupe M3².

² Par exemple: Une étudiante a apporté des papillotes la semaine suivante.

3.4 Conclusions et perspectives.

3.4.1 La friandise.

L'effet de cette action sur les étudiants est très subjectif. Cependant, il semble qu'un effet bénéfique soit visible. L'ambiance générale et les relations avec l'enseignant en particulier s'en trouvent améliorées. Cette technique est utilisable à peu de frais (pas de temps perdu) pour peu que l'enseignant soit sincère et sache amener naturellement la chose.

3.4.2 Le remue-méninges.

Le remue-méninges a eu un effet très positif sur le groupe M3: motivation, dynamisme, plaisir évident d'apprendre. Cet effet fut moindre dans le groupe M4. La différence s'explique peut-être par le fait que les attentes des étudiants au départ n'étaient pas les mêmes. Dans le groupe M3, le T.D. était obligatoire, il y avait peu d'attentes concrètes et tout effort pédagogique original était le bienvenu. Dans le groupe M4, les attentes et les demandes étaient fortes, la présence volontaire et on sentait le désir de profiter le plus possible des micro-ordinateurs. Le remue-méninges pouvait être alors considéré comme un temps pendant lequel les ordinateurs n'étaient pas utilisés.

Cette technique a été globalement appréciée, voire réclamée par les étudiants. Même si le but n'était pas qu'ils apprécient l'exercice, il ne faut pas négliger le plaisir qu'ils semblaient en tirer. D'autant que cet exercice nécessite beaucoup d'énergie et de participation. Seules deux étudiantes en dans le groupe M4 ont émis l'opinion que ce n'était pas efficace, et qu'il fallait peut-être arrêter, ou le faire plus vite. Les autres se sont immédiatement portés en faux, ce qui a été un encouragement à poursuivre dans cette voie.

3.4.3 Perspectives.

En résumé, cette expérience a été globalement satisfaisante bien que certains points puissent être améliorés. En particulier, il faudrait trouver comment faire participer davantage les étudiants qui ont du mal à s'exprimer devant leurs camarades. une idée: faire le remue-méninges en demi-groupe, avec un étudiant dans le rôle d'observateur .

Ce type d'exercice peut être recommandé dans les groupes d'étudiants peu motivés à suivre un cours obligatoire. Pour les étudiants volontaires, il est plus sûr de ne pas le faire, et de profiter le plus possible de la disponibilité des micro-ordinateurs. Il est intéressant aussi pour commencer un T.D. tôt le matin car il permet de mobiliser l'attention des étudiants.

Du point de vue de l'enseignant, c'est une surcharge de travail: il faut préparer soigneusement, être attentif, dépouiller ces notes, et assurer le même enseignement avec 10 minutes en moins. En se basant sur la motivation exprimée, l'assiduité en libre service ainsi que le nombre de fiches techniques réalisées, ces 10 minutes ne sont pas du temps perdu.

Enfin, une évaluation plus complète et objective, avec des données fiables pour d'autres groupes animés par le même enseignant serait indispensable. L'enseignement est beaucoup affaire de contacts humains, très variables d'un individu à l'autre. Il faudrait un échantillon plus large pour obtenir des statistiques plus significatives. Cette évaluation pourrait être complet(e) par une évaluation par l'étudiant lui-même de l'impact de ces techniques, et par une vérification du contenu des notes prises par les étudiants pendant le TD.

4 Conclusion.

Dans cette modeste étude, une expérience d'enseignement a été rapportée. Elle vise à favoriser par des techniques basées sur des modèles théoriques de la relation enseignant - étudiant et du processus d'apprentissage l'assimilation d'un T.D. d'informatique. Les résultats sur trois groupes d'étudiants, répartis sur deux années sont encourageants mais doivent être complétés par une étude plus globale et plus rigoureuse. Déjà, quelques éléments intéressants ont été mis à jour; par exemple que la technique du remue-méninges semble être particulièrement bénéfique pour des groupes d'étudiants ayant de faibles motivations. Nous espérons être en mesure de poursuivre ces expériences dans les meilleures conditions possibles. Pour conclure, ces techniques ne sont pas spécifiques à l'enseignement de l'informatique: elles peuvent être appliquées a priori dans tout autre domaine.

30 Juin 1995

Patrice De Marconnay

LIFIA - IMAG, 46 av. F. Viallet,

38031 Grenoble, France

Tél: (33) 76-57-46-09

Fax: (33) 76-57-46-03

email: Patrice.De-Marconnay@imag.fr

A - Annexes.

A.1 Détail de quelques formations du C.I.E.S. de Grenoble.

- F3 - Techniques de communication I.

- Organisation: C. I.E.S.
- Contenu: exercices de communication, apprentissage de techniques de travail interactives et créatrices, communication inter-personnelle, communication de groupe.
- Intervenants: Mr H. Raynaud.
- Durée: 3 jours.
- Nombre de participants: 13.
- Principaux intérêts personnels: prise de conscience du rôle prédominant de la communication, de l'importance de l'aspect relationnel dans l'enseignement, de l'existence de techniques originales de transfert de connaissances (brains-torming).

- F5 - Techniques de communication II.

- Organisation: C.I.E.S..
- Contenu: bilan individuel du premier stage, les rôles de l'enseignant, les phénomènes de groupe, le groupe classe, la modification de comportements, la gestion du temps.
- Intervenants: Mr H. Raynaud.
- Durée: 3 jours.
- Nombre de participants: 10.
- Principaux intérêts personnels: observation des phénomènes de groupe, recul par rapport à l'idée que bien enseigner, c'est d'abord bien être (d'où l'utilité des techniques de changement de comportements), et aussi être soi.

- F8 - Techniques de communication III.

- Organisation: cette formation a été réclamée par une dizaine de moniteurs, organisée et financée par eux.
- Contenu: une expérience forte des phénomènes psychologiques sous-jacents à la naissance, à la vie et à la mort d'un groupe (training-group).
- Intervenants: Mr H. Raynaud.
- Durée: 3 jours.
- Nombre de participants: 11.
- Principaux intérêts personnels: une expérience personnelle d'une grande richesse, une compréhension vécue des phénomènes de groupe, le début de relations d'amitié fortes avec d'autres moniteurs.

A.2 Détail des enseignements.

- M2 - T.D. DOS - Word 5.

- Type : T.D. obligatoire
- Objectifs: apprendre aux étudiants à utiliser un micro-ordinateur; leur donner quelques notions de base en informatique (fichier, répertoire, stockage et présentation de données, système d'exploitation, imprimantes) et en traitement de textes (document, mise en forme, style, ...).
- Contenu: généralités sur l'informatique; DOS; Word 5.
- Profil des étudiants: Deug L.E.A. 1ère année.
- Nombre d'étudiants: un groupe de 16, un autre de 15.
- Remarque sur les étudiants: motivation et niveau moyens.
- Période: le semestre 93-94
- Horaire: mardi 13h30 -15h30 et jeudi 8h30-10h30.
- Volume horaire: 56 h (deux fois 13 séances + examens).
- Durée: 2 h.
- Conditions matérielles: une salle de 8 compatibles 386 avec écrans couleur, une autre de 286 avec écrans monochromes.
- Autres enseignements associés: un C.M. sur l'informatique.
- Responsable de l'enseignement: Mme M. H. Stéfani.
- Examen: création, surveillance et correction d'un examen pratique.

- Principales difficultés: il fallait s'adapter à un public plutôt orienté langues et lettres, qui n'a pas les mêmes références et difficultés que moi. Ainsi, certaines notions que je pensais faciles avaient du mal à être assimilées, d'autres selon moi difficiles ne posaient aucun problème. Le réajustement ne s'est pas fait sans peine. Le groupe du jeudi matin (8h30) était dur à réveiller.

- Points intéressants: réflexions passionnantes sur le contenu et surtout la forme du T.D.: le support (tableau, transparent, écrans), l'enchaînement de la théorie et de la pratique, que faire pour motiver, etc.

- M3 - T.D. Windows- WinWord.

- Type: T.D. obligatoire.

- Objectifs: les mêmes que M2.

- Contenu: généralités sur l'informatique; Windows; Word sous Windows.

- Profil des étudiants: Deug Sciences du Langage 1ère année.

- Nombre d'étudiants: 17.

- Remarque sur les étudiants: peu motivés au départ; dynamiques par la suite.

- Période: 1er semestre 94-95.

- Horaire: mercredi 13h30- 15h30.

- Volume horaire: 24 h. (11 séances + examen).

- Durée: 2 h

- Conditions matérielles: les mêmes que M1.

- Autres enseignements associés: un C.M. sur l'informatique.

- Responsable de l'enseignement: Mr G. Antoniadis.

- Examen: création, surveillance et correction d'un examen pratique.

- Principales difficultés: le premier T.D. était un peu tendu: peu d'étudiants étaient motivés (cours obligatoire), certains étaient sur la défensive.

- Points intéressants: certaines techniques de communication peu utilisées habituellement ont bien rempli leur rôle.

- M4- T.D. Windows- WinWord.

- Type: T.D. optionnel.

- Objectifs: les mêmes que M2.

- Contenu: le même que M3.

- Profil des étudiants: I.P.T.³, de Deug 1ère année à Maîtrise.

- Nombre d'étudiants: 17.

- Remarque sur les étudiants: beaucoup d'attentes personnelles.

- Période: 1er semestre 94-95.

- Horaire: mercredi 8h30- 10h30.

- Volume horaire: 24 h. (11 séances + examen).

- Durée: 2 h.

- Conditions matérielles: les mêmes que M1.

- Autres enseignements associés: un C.M. sur l'informatique.

- Responsable de l'enseignement: Mme V. Guéraud.

³ Informatique Pour Tous.

- Examen: création, surveillance et correction d'un examen pratique.
- Principales difficultés: attentes et niveaux très hétérogènes; débuts de T.D. toujours un peu fastidieux.
- Points intéressants: beaucoup de questions et une grande exigence: l'informatique est pour eux un outil dont ils ont besoin.

A.3 Exemples de fiches techniques.

LE GESTIONNAIRE D'IMPRESSION

1. Son rôle

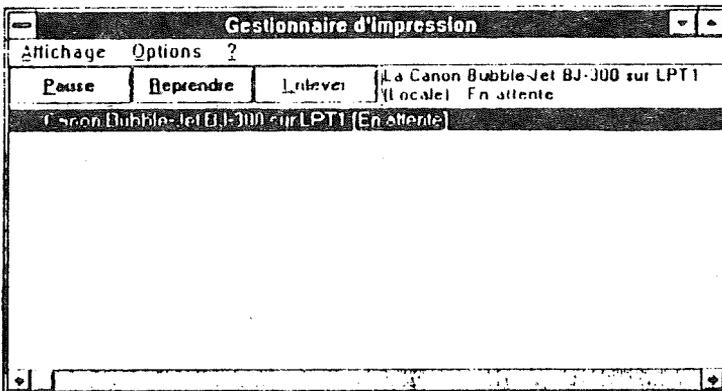
Le gestionnaire d'impression permet avant tout de gérer les documents envoyés à une imprimante. Mais il travaille en arrière plan et l'utilisateur n'y a donc recours qu'en cas de problèmes lors de l'impression d'un document (imprimante défaillante, etc ...). Il permet donc à l'utilisateur de continuer de travailler sur une autre application durant l'impression.

2. Lancement

Pour lancer le gestionnaire d'impression, il suffit de se rendre dans le fichier de l'application en cours et de double-cliquer sur "imprimer".

Le gestionnaire d'impression crée alors un fichier d'impression qui est directement envoyé à l'imprimante.

Cependant on peut se rendre dans l'application du gestionnaire d'impression en passant par le groupe principal et en double-cliquant sur son icône.



Ceci est utile lors de:

- Changement d'ordre dans la file d'attente
- Modification du titre du document
- Interruption et reprise de l'impression (cliquer sur "pause" et "reprendre")
- Annulation de l'impression d'un document (cliquer sur "enlever")
- Sélection de l'imprimante par défaut

3. Comment quitter le gestionnaire d'impression ?

Lorsque l'impression des documents en attente est terminée, la fermeture du gestionnaire d'impression se fait automatiquement.

Si vous êtes dans l'application du gestionnaire d'impression, il suffit de double-cliquer sur la case "menu-système" ou de cliquer sur enlever.

FICHE MEMOIRE

21 Décembre 1994.

ACCESSOIRE DE WINDOWS - L'HORLOGE

I .DEFINITION

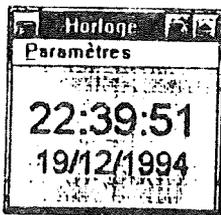
L'horloge permet de vérifier l'heure à tout instant.

II . ACCES A L'HORLOGE

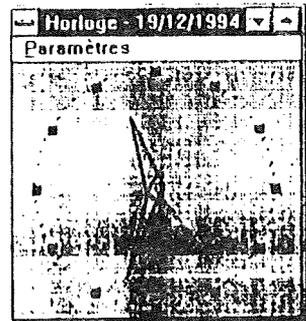
- Gestionnaire de Programmes
- Accessoires
- Horloge (double clique).

III. PRESENTATION.

A. Numérique.



B . Analogique.



C-Variantes dans la présentation**-Possibilité de la faire apparaître en permanence à l'écran.**

Démarche à suivre: Nécessité de faire apparaître l'horloge (cf 11°)
Cliquer sur la barre du menu horloge
Option: Toujours visible.

- Changement police

Démarche à suivre: Nécessité de faire apparaître l'horloge (cf 11°)
Paramètres
Option: Police...

-Diverses: Apparition ou non de la Date et des Secondes.

Démarche à suivre: Apparition de l'horloge (cf 11°)
Paramètres
Options: - Date
- Secondes

TD Informatique

**Thème4 : Informatique au Supérieur
(Sciences Techniques)**

Au ras des paquerettes

Guy CHATTY

présenté le jeudi 11 avril 1996 à 15h45

Résumé: Nous montrons, sur quelques exemples, qu'on peut aider à l'initiation aux algorithmes élémentaires de base en les exprimant sous forme de sentences proverbiales correspondant à leur sens. Ceci permet un regroupement des algorithmes par familles, facilite la compréhension ainsi que la mémorisation de ces algorithmes, et n'empêche pas leur approfondissement mathématique. De plus, pour certains élèves, ce peut être une préparation à l'étude des schémas d'algorithmes.

1. Introduction

Il est des circonstances où l'on doit initier à l'algorithmique un public hétérogène composé en partie de personnes ayant une culture mathématique faible.

Pour faciliter la compréhension de ces dernières, on est poussé parfois dans les derniers retranchements de tentatives d'explication simple. On finit par condenser des analyses en des formulations qui s'apparentent davantage à des proverbes contenant un certain bon sens qu'à des processus mathématiques. Et on s'aperçoit que cela ne manque pas d'intérêt d'autant plus qu'on peut établir un lien étroit entre ces formulations et la preuve mathématique des algorithmes concernés, et que rien ne nous empêche d'exposer celle-ci dans toute sa rigueur. Au bout du compte, on a aidé à la fois à la compréhension du processus algorithmique et du support mathématique.

De plus, ces formulations mettent en évidence ce qu'il peut y avoir de commun dans plusieurs algorithmes et préparent ainsi le regroupement en catégories de certains algorithmes d'où une économie de pensée et une aide à la mémorisation et à la recherche des algorithmes. Elles débouchent d'ailleurs parfois sur les schémas d'algorithmes [2] dont l'étude n'est pas envisagée ici.

Nous donnons ici quelques exemples de ces formulations. Les algorithmes étudiés sont des algorithmes élémentaires de base [1].

Nous utilisons un pseudo-langage d'expression des algorithmes possédant les primitives classiques: opérations arithmétiques, affectation, les primitives de choix: *si alors, si alors sinon*, les primitives d'itération: *tant que, répéter jusqu'à, pour*.

2. "Si c'est mieux, on le prend"

Considérons le problème de la recherche du plus petit élément, ou du plus grand, dans une suite (u_i) de n éléments d'un ensemble A . On suppose que A est totalement ordonné par une relation notée $<$.

Pour le résoudre avec les primitives classiques, l'idée est de définir une variable X initialisée à u_1 et de parcourir toute la suite du début à la fin, en ne modifiant à chaque étape la valeur de X que si l'élément courant est "meilleur" (plus petit ou plus grand) que cette valeur. L'essentiel de l'algorithme est donc (on a choisi l'option "plus petit") :

```

X := u1
tant que i < n faire
    i := i + 1
    si ui < X
        alors X := ui
    fin si
fin tant que
  
```

L'idée à souligner, et qui n'est pas toujours vite et bien comprise, est que dans l'autre cas ($u_i > X$), on ne fait rien. D'où la formulation "Si c'est mieux, on le prend".

La preuve mathématique de cet algorithme repose sur le raisonnement par récurrence suivant :

Si $p(i)$ représente le minimum de l'ensemble des i premiers éléments de la suite, c'est-à-dire $p(i) = \min \{u_1, u_2, \dots, u_i\}$,
alors $p(i + 1) = \min \{u_{i+1}, p(i)\}$.

Théorème facile à démontrer, mais demandant cependant quelque attention.

Le minimum d'une suite de 1 élément étant cet élément lui-même, on peut mettre en évidence une suite récurrente dont le dernier terme est la solution du problème.

$$\begin{aligned}
 p(1) &= u_1 \\
 p(2) &= \min\{u_2, p(1)\} \\
 &\dots \\
 p(n-1) &= \min\{u_{n-1}, p(n-2)\} \\
 p(n) &= \min\{u_n, p(n-1)\}
 \end{aligned}$$

Par conséquent, pour calculer le minimum p de la suite (u_i) de n éléments, il faut calculer les éléments successifs de la suite récurrente $(p(i))$. Ce que l'on peut exprimer par l'algorithme ci-dessus, la variable X recevant les valeurs $p(i)$, i variant

de 1 à n : la nouvelle valeur de X à l'étape i+1 est bien le minimum entre l'ancienne et u_i . L'algorithme ne fait que mettre en œuvre la suite (p(i)).

Notre formulation nous a quand même ouvert la voie !

3. "Tant que ce n'est pas fini, on avance".

Considérons les circonstances où l'on veut détecter ou insérer, dans une suite de n éléments d'un ensemble A, un élément de A ou une sous-suite d'éléments de A, vérifiant une condition.

Elles se présentent dans les problèmes suivants et peut donner lieu à différentes formulations.

a) "Tant qu'il y a de l'espoir, on avance".

- **Problème.** Rechercher un élément X dans une suite quelconque.

On avance dans la suite tant que X est différent de l'élément visité et qu'on n'est pas arrivé à la fin.

```
i:=1
tant que X_ ui et i<n faire
    i:=i+1
fin tant que
trouvé:=X=ui
```

trouvé est une variable booléenne de valeur vraie ou fausse.

- **Problème.** Rechercher un élément X dans une suite triée, dans "l'ordre croissant" par exemple.

On avance dans la suite tant que X est supérieur à l'élément visité et qu'on n'est pas arrivé à la fin.

```
i:=1
tant que X> ui et i<n faire
    i:=i+1
fin tant que
trouvé:=X=ui
```

Lorsque X est inférieur ou égal à u_i , soit il n'y a plus d'espoir dans le premier cas, soit on a trouvé X : dans les deux cas, on sort de la boucle.

b) "Tant que c'est bon, on avance".

- **Problème.** Vérifier si une chaîne de caractères de longueur n représente un entier. On avance dans la suite tant que le caractère est un chiffre et qu'on n'est pas arrivé à la fin.

```

i:=1
tant que ui est un chiffre et i2n faire
    i:=i+1
fin tant que
chaîne-entier:=i=n+1

```

- **Problème.** Insérer un élément "à sa place" dans une suite triée dans "l'ordre croissant" par exemple.

On avance dans la suite tant que l'élément à insérer est supérieur à l'élément visité et qu'on n'est pas arrivé à la fin.

```

i:=1
tant que E > ui et i2n faire
    i:=i+1
fin tant que

```

A la sortie de la boucle, i donne la place cherchée.

- **Problème.** Trouver une monotonie (sous-suite maximale triée) dans une suite quelconque.

On avance dans la suite tant que l'élément visité est inférieur à son suivant et qu'on n'est pas arrivé à la fin.

```

i:=1
tant que ui2 ui+1 et i < n-1 faire
    i:=i+1
fin tant que

```

A la sortie de la boucle, i donne le rang du dernier terme de la monotonie.

Ces deux formulations "*Tant qu'il y a de l'espoir, on avance*" et "*Tant que c'est bon, on avance*" peuvent être rassemblées en une seule : "*Tant que ce n'est pas fini, on avance*", le "*ce n'est pas fini*" correspondant à une condition attachée au problème qui est encore vérifiée, ou ne l'est plus, à l'étape considérée.

On tombe sur un schéma d'algorithme mais il n'est pas question ici d'aller du général au particulier : au contraire, la démarche est de faire prendre conscience intuitivement sur des cas précis du sens de l'algorithme et de regrouper ensuite ce qui peut être regroupé. Notre propos est "au ras des pâquerettes" pour mieux préparer "l'envol".

4. "Tant que ce n'est pas fini, on calcule et on avance"

Dans les exemples précédents, la suite était donnée. On peut être amené à la construire. Il en est ainsi dans le calcul approché de Va par la méthode de Newton.

On calcule les éléments de la suite (x_i) :

$$x_i = 1/2 (x_{i-1} + a/x_{i-1}) \text{ si } i > 0$$

$$x_0 = a/2$$

tant que $|x_i - x_{i-1}| / x_{i-1} > \epsilon$.

Ici, le "on avance" s'agrément de calculs qui permettent de calculer le terme suivant de la suite.

```

algorithme racine
objets A, _, X, Y
début
lire (A, _)
X:=A/2
Z:=1
tant que Z > _ faire
    Y:=(X+A/X)/2
    Z:=|Y-X| / X
    X:=Y
fin tant que
écrire (X)
    
```

Montrer la *finitude* de l'algorithme est chose aisée avec tous les exemples précédents. Cela peut être plus délicat avec l'application du proverbe "*Tant que ce n'est pas fini, on avance ou on recule*" !

Pour l'algorithme suivant, où n est un entier naturel, on démontre facilement que toute exécution a une fin :

```

début
lire (n)
tant que n _ 1 faire
    si n est pair
        alors n := n/2
        sinon n := 2n + 1
    fin si
fin tant que
fin
    
```

Par contre, pour le suivant, on ne sait pas encore le démontrer !

```

début
lire (n)
tant que n _ 1 faire
    si n est pair
        alors n := n/2
        sinon n := 3n + 1
    fin si
    
```

fin tant que
fin

5. "Tant que ce n'est pas fini, on avance"...dans une arborescence "

On se place ici dans le cas du traitement d'une arborescence. La formulation "Tant que ce n'est pas fini, on avance" apparaît par exemple dans le problème suivant.

Problème. Tri arborescent d'une suite

Construction de l'arborescence binaire A

Analyse. On construit la racine et on y associe le premier élément. D'une manière générale, tant que l'élément considéré U n'est pas placé, on avance dans l'arborescence déjà construite, à gauche ou à droite du sommet S suivant que U est plus petit ou plus grand que p(S), élément associé à S. Si la place est libre, on construit le nouveau sommet et on lui associe U.

FINI, variable booléenne initialisée à faux, prendra la valeur vrai quand U sera placé.

```

répéter
    si U < p(S) alors
        si libre à gauche
            alors
                placer U à gauche
                FINI:=vrai
            sinon
                aller à gauche
        finsi
    sinon
        si libre à droite...
        finsi
    finsi
jusqu'à FINI
  
```

6. "A chaque étape, on agit s'il le faut".

Le proverbe "Si c'est mieux, on le prend" du §1 est déjà un exemple de "A chaque étape, on agit s'il le faut".

Donnons d'autres exemples.

- **Problème.** Chercher les diviseurs d'un entier en est un :

début
lire(n)

```

répéter
    si  $n \bmod d = 0$ 
        alors écrire "d est un diviseur de n"
    fin si
d:=d+1
jusqu'à  $d > n$ 
    
```

- Dans le "tri à bulles", on utilise :

```

pour k de 1 à i-1 faire
    si  $u_k > u_{k+1}$ 
        alors échanger ( $u_k, u_{k+1}$ )
    fin si
fin pour
    
```

7. "Tant qu'il y a eu modification, recommencer"

La formulation "Tant qu'il y a eu modification, recommencer" apparaît dans les cas où la fin d'itération a lieu lorsqu'au précédent passage dans la boucle, les instructions de traitement n'ont pas été exécutées.

Ces cas se présentent dans des problèmes très différents à première vue.

Problème : Tri à bulles

Analyse (extrait)

Pour sortir de la boucle "parcours et échanges éventuels", on teste s'il y a eu effectivement échange au dernier passage : on peut utiliser une variable booléenne E initialisée à faux dans la boucle, et qui prend la valeur vraie en cas d'échange.

```

E:=vrai
Tant que E faire
    E:=faux
    {parcours et échanges éventuels}
    {E prend la valeur vrai en cas d'échange}
fin tant que
    
```

Problème : Recherche des composantes fortement connexes (cfc) d'un graphe.

Méthode : marquer + les descendants et - les ascendants d'un sommet x. Les sommets marqués + et - constituent la cfc de x.

Pour le marquage, il faut faire un balayage des sommets non encore "pris".

Il est nécessaire, pour terminer, de faire un balayage "blanc" où aucun sommet n'est marqué.

On peut utiliser une variable booléenne M initialisée à faux dans la boucle et qui prend la valeur vraie en cas de marquage.

```

M:=vrai
Tant que M faire
    M:=faux
    {balayage}
    {M prend la valeur vrai en cas de marquage}
fin tant que
  
```

8. "Pour comparer, sauvegarder".

a) Les doublons

Considérons le problème suivant .

Problème Lire un texte caractère par caractère (la fin est indiquée par *), compter le nombre de couples de lettres consécutives identiques (on suppose que le texte contient au moins un caractère différent de * et qu'une lettre ne se répète pas à la suite plus d'une fois), écrire ce nombre et la longueur du texte.

Analyse.

On lit le texte caractère par caractère (C) jusqu'à rencontrer le caractère *. En même temps, pour chaque caractère à partir du deuxième, on regarde s'il est identique au précédent (P). Celui-ci doit donc être conservé ! Si C=P , on incrémente un compteur (NC). Pour calculer la longueur, on utilise un autre compteur (I).

Algorithme.

```

algorithme LIRTEXT
début
NC := 0
lire (C)
I := 1
répéter
    P := C { la valeur de C est sauvegardée dans P}
    lire (C) { le nouveau caractère est placé dans C}
    si C = P alors NC := NC + 1
    fin si
    I := I + 1
jusque C = '*'
écrire ( NC, I)
fin
  
```

b) Tri d'un fichier en organisation séquentielle

Un autre exemple de ce proverbe "Pour comparer, sauvegarder" est donné dans le tri d'un fichier séquentiel F, lorsqu'on répartit les monotonies de F sur deux fichiers FA et FB puis qu'on fusionne 2 à 2 les monotonies de FA et de FB sur le fichier F (qui ne sont pas nécessairement les monotonies obtenues dans la

répartition). Pour détecter la fin de ces monotonies, on est amené à utiliser deux fenêtres pour chacun des fichiers FA et FB et à comparer leurs contenus.

9. "Là où l'on prend, là on avance".

Considérons le problème de la fusion de deux suites triées (a_i) , $1 \leq i \leq n$, et (b_j) , $1 \leq j \leq m$, en une suite triée (c_k) dont l'algorithme peut s'écrire (on a complété les suites par deux termes plus grands que les termes des deux suites pour éviter de vider une suite quand l'autre est terminée) :

```

début
i := 1; j := 1; k := 1;
répéter
    Si  $a_i \leq b_j$ 
        alors
             $c_k := a_i$ 
            i := i + 1
        sinon
             $c_k := b_j$ 
            j := j + 1
    Fin Si
    k := k + 1
jusqu'à FINSUITES

fin
    
```

Pour étendre l'algorithme à la fusion de deux fichiers, on peut remplacer les incréments du type $i:=i+1$ par "avancer dans (u_i) ".

On a la formulation qui résume ces algorithmes : "là où l'on prend, là on avance".

10. Conclusion

Nous avons montré sur quelques exemples qu'on pouvait aider à l'initiation aux algorithmes élémentaires de base en les exprimant sous forme de sentences proverbiales correspondant à leur sens.

Ceci permet un **regroupement des algorithmes** par familles, facilite la **compréhension** ainsi que la **mémorisation** de ces algorithmes, et n'empêche pas leur approfondissement mathématique.

De plus, pour certains étudiants, c'est une préparation à l'étude des schémas d'algorithmes. Enfin, les étudiants eux-mêmes peuvent se prendre au jeu et inventer à leur tour des proverbes, ce qui va dans le sens de l'objectif poursuivi et ne peut que réjouir le professeur.

Pour les élèves de l'enseignement secondaire et les personnes étrangères au monde informatique, les plus simples de ces formulations peuvent donner une

première idée des processus algorithmiques et constituer des éléments d'une culture générale informatique.

Guy CHATTY

Université Paris Nord - IUT de Villetaneuse
Avenue Jean Baptiste Clément, 93430 Villetaneuse - France
Tél.: (1) 49 40 31 48 fax: (1) 48 26 30 59

Références.

- [1] G.Chaty et J.Vicard, Programmation, Ellipses, Paris 1992.
- [2] P-C. Scholl et J.-P. Peyrin, Schéma algorithmiques fondamentaux, Masson, Paris 1989.

Une expérience d'enseignement optionnel de Caml à l'Université

Daniel BOURGET

présenté le vendredi 12 avril 1996 à 11h00

I- Préliminaire

Quelque soit la discipline considérée, l'introduction d'un nouvel enseignement dans un cycle donné d'enseignement répond en général à des préoccupations d'apprentissage de toute, ou d'une partie de toute une production scientifique récente ayant fait ses preuves, et ceci avec un soucis de cohérence dans un profil de formation. Il va de soit qu'ensuite se pose la nécessité de l'existence, ou sinon de la création de toute une logistique pédagogique et d'un encadrement humain de qualité.

Au département informatique de l'université de Brest, s'est manifestée la volonté de tirer profit des premières expériences internationales d'enseignement d'un langage de la classe ML [Mil83], et des possibilités qu'il offre en tant que première approche d'un grand nombre de concepts liés, soit à l'informatique "applicative", soit à l'informatique tout court (par exemple, la notion de fonction et de programme, la récursivité, les types, la modularité, l'encapsulation).

Il a donc été décidé d'introduire dans un module optionnel d'informatique en Deug A2, un enseignement de Caml. Cette décision a été prise suite aux possibilités:

-- d'utiliser un matériel adéquat (10 stations Sun) à Brest et 24 PC/386 à Lorient.

-- d'avoir accès à un logiciel Caml-light [Ler92,Ler93,Wei93] en Freeware qui globalement répond correctement aux nécessités d'enseignement pratique tel que nous l'avons envisagé, et aussi à des contacts réguliers avec le "monde Caml" en France (précédente réunion sur l'enseignement des langages applicatifs, enseignants du CNAM] de Paris [2], etc).

L'enseignement de Caml [Mau92] s'est adressé à un public d'étudiants inscrits à Brest et à Lorient pendant le second semestre avec la répartition suivante:

- 9 heures de cours ;
- 8 heures de travaux dirigés ;
- 8 heures de travaux pratiques.

Les étudiants avaient reçu l'année précédente, et au cours du premier semestre de la même année, un enseignement de{PASCAL} (langage et éléments d'algorithmique). Une grande partie d'entre eux avaient acquis, soit au lycée, soit

par des moyens personnels, des connaissances complémentaires assez disparates sur d'autres langages de programmation BASIC, C, LISP, assembleur).

Un contrôle de connaissances a eu lieu en fin d'année, avec des résultats satisfaisants pour la majorité des étudiants, tant à l'épreuve écrite qu'à l'épreuve pratique.

Un formulaire d'enquête anonyme a été distribué aux étudiants après la fin des enseignements, la moitié des étudiants y ont répondu. A travers ce questionnaire, nous avons tenu à réunir des renseignements sur les formations antérieures des étudiants, et des éléments d'appréciation sur la perception et la compréhension des concepts véhiculés par le langage Caml (la modularité, l'encapsulation, la curryfication et le typage), ainsi que sur l'acquis en nouvelles connaissances et sur l'intérêt de cet enseignement.

Dans ce qui suit nous présentons les enseignements tels qu'ils ont été effectués avec quelques remarques.

II- Le Cours

Le cours [Bou93] a été réparti sur 9 séances de 1 heure. La première séance a mis en avant les insuffisances du langage Pascal en terme de programmation et surtout en terme de génie logiciel. Les étudiants dans leur ensemble ont été surpris par cette approche de l'informatique. En fait, ils avaient une vue très restreinte des concepts de la programmation et du génie logiciel qui semble pourtant particulièrement leur convenir.

Nous avons voulu, en choisissant le langage Caml, élargir les domaines de la programmation et du génie logiciel accessibles pour l'étudiant..

Les notions de styles de programmation et l'introduction du génie logiciel moderne ont été les bases de notre formation. Nous avons mis dans certaines situations les étudiants fassent à leur responsabilités en tant que programmeur. Nombreux sont ceux qui ont compris que l'on respecte certaines règles dans le monde industriel.

Les étudiants dans leur ensemble ont été très surpris sur la qualité du logiciel qu'ils produisaient si ils respectaient un certain nombres de concepts.

Des notions comme le typage, la compilation séparée, la généricité, la création et l'utilisation de composants logiciels, nous sont apparues fondamentales. Il est à noter que ce sont ces notions qui nous ont orientées vers Caml.

Le typage est particulièrement intéressant, qu'il s'agisse de typage statique ou du typage dynamique. Caml] est un très bon outil pour une approche simple et suffisamment complète de l'informatique en général illustrant aussi bien les concepts des langages applicatifs que ceux des langages impératifs. L'aspect

fonctionnel a été très développé au détriment de l'aspect impératif car les étudiants, avec la formation Pascal, ont eu une première approche de la programmation impérative. Les notions conceptuelles de type, de variable, d'affectation ont été acquises ainsi que les mécanismes de boucles et de structures de contrôles. Par conséquent, seul le dernier cours a porté sur les aspects impératifs du langage Caml, de plus, comme ce langage combine la programmation applicative et fonctionnelle, nous avons pu montrer aux étudiants la capacité d'abstraction entre un programme pascal et sa version Caml d'un côté et une implémentation purement fonctionnelle d'un autre côté. Par exemple, le célèbre problème des huit reines est présenté suivant deux philosophies différentes. L'une impérative, avec un ensemble de boucles et de variables d'affectation et une autre purement fonctionnelle, type flot de données. Les étudiants ont pu discerner les différences au niveau de la résolution du problème en vue de son implémentation dans un langage informatique selon que l'on utilise un modèle de raisonnement impératif ou fonctionnel.

Le mécanisme de typage de fonctions que celles-ci soient simples ou polymorphes a été difficile à faire assimiler. De même que les concepts de curryfication et décurryfication, sans doute à cause du manque de temps. Des exercices de typage ont été régulièrement proposés en travaux dirigés de façon à ce que les étudiants se familiarisent avec la technique du typage. Cette démarche a été retenue dans chaque cours.

Le principe de filtrage a été très bien assimilé et utilisé systématiquement par les étudiants. Cela leur semble proche de la définition de fonctions mathématiques définies par morceaux. Quant aux fonctions du langage Lisp, car, cdr, ..., bien que leurs équivalents existent en Caml, elles n'ont pas été utilisées en travaux dirigés, mais présentées en cours.

La récursivité simple, croisée et terminale n'a pas posé de réelles difficultés sauf au niveau du typage.

L'introduction de la gestion des exceptions a permis aux étudiants d'entrer dans un monde nouveau. Le fait de concevoir des programmes en gérant totalement les erreurs a motivé les étudiants plutôt que de les déstabiliser.

L'étude du type liste d'une part, et du type somme d'autre part, a permis d'introduire les premières structures de données non triviales comme les arbres binaires ou n-aires.

La compilation séparée et la conception de composants logiciels avec les mécanismes d'encapsulation des données a donné aux étudiants une idée sur la façon de concevoir un logiciel dans le monde de la production industrielle.

Il est à noter que deux expériences d'enseignement Caml ont été menées parallèlement à l'université de Brest et au centre universitaire de Lorient. La

différence s'est faite essentiellement sur les concepts de programmation séparée, sur la construction de composants logiciels, sur l'encapsulation et la réutilisabilité de ces composants et sur l'introduction du style impératif. A Brest ces notions ont été introduites à la fin du cours, en revanche à Lorient, ces concepts ont fait l'objet d'études approfondies. Aussi, la différence entre les deux auditoires pour l'appréhension de ces problèmes a été sensible. La deuxième population a mieux acquis ces différents concepts.

Il est à souligner que la construction de composants logiciels a été considérée comme quelque chose de naturelle en séparant les spécifications du corps des différentes fonctions constituant un module. L'encapsulation des structures de données des différents composants d'un module a été bien assimilée.

III- Les travaux dirigés

Les TDs ont été répartis sur 4 séances de 2H, soit au total 8H. La première séance a été précédée d'une séance de Travaux pratiques, ce qui a permis aux étudiants de se familiariser avec l'aspect fonctionnel de Caml.

Les exercices proposés avaient pour principal objectif l'assimilation du langage Caml et des concepts véhiculés à travers lui. Aussi, diverses notions ont été abordées:

-- La technique de typage s'est avérée un outil très apprécié en général pour l'aide à l'écriture et à la lecture des programmes Caml. Certaines difficultés sont apparues à ce niveau, surtout pour typer les fonctions contenant des instructions impératives. On peut en déduire que le typage devient plus complexe avec une syntaxe non fonctionnelle. Le typage des fonctions contenant des variables polymorphes est également difficile en général à appréhender au début, mais nous pensons qu'avec plus de pratique ceci devient aisé ;

-- Les exercices ont couvert divers types, : int, float, bool, exception, somme, etc ... Le type liste a fait l'objet d'une étude approfondie, ce qui a permis d'expliquer aisément des notions telles que le filtrage et le polymorphisme. Le filtrage a aidé les étudiants à écrire des fonctions très concises et très proches de la spécification des fonctions mathématiques. Le type exception a été utilisé principalement pour définir des fonctions totales à partir de fonctions partiellement définies, ce qui est le cas de nombreuses fonctions en Caml. Il a servi également dans la récupération des erreurs. Ce dernier aspect a fait l'objet de nombreuses interrogations sur la gestion des erreurs dans le langage Pascal, qui est le langage connu des étudiants.

-- Au niveau algorithmique, parmi les algorithmes usuels, nous citerons: es tris et plus particulièrement le Quick Sort. La génération du code Caml de la fonction réalisant ce tri a été très facile. Il s'est avéré beaucoup plus aisé d'expliquer cet algorithme, en comparaison avec le même écrit en Pascal.

D'autres algorithmes numériques ont été utilisés, : Newton, génération des nombres premiers, manipulation des rationnels, etc

La récursivité qui est une nouvelle notion pour les étudiants, a été largement étudiée et plus particulièrement la récursivité terminale. Ce dernier aspect a posé quelques problèmes au niveau du typage. Et en plus, cette démarche paraît anormale contrairement à la récursivité standard.

-- Au niveau de l'écriture des programmes, le parenthésage a parfois posé quelques problèmes aux étudiants. Exemple

```
let rec pgcd a b =  
  if b = 0  
  then a  
  else pgcd b a mod b
```

Le problème est que le manque de parenthèses autour de `a mod b` est non détectable à la compilation.

-- Au niveau génie logiciel les exercices ont permis d'étudier la programmation dirigée par les données aboutissant ainsi à des programmes très courts et facilement modifiables.

La modularité par compilation séparée est une nouvelle approche de programmation pour les étudiants. La facilité de ce procédé en Caml aide à écrire des modules fonctionnels clairement définis. Les étudiants se sont spontanément posés le problème de la surcharge.

En dernier lieu certains types abstraits ont été étudiés à travers l'écriture de fonctions manipulant les rationnels.

IV- Les travaux pratiques

Les étudiants ont travaillé sous Unix, avec Sun/Os. Un essai mené préalablement, dans le cadre de l'enseignement de Pascal, avait mis en évidence qu'il était déraisonnable de quitter l'environnement douillet des compatibles PC sans proposer à l'élève un confort au moins équivalent ; il ne pouvait être question de placer nos étudiants sur des TTY. Un autre problème se posait : le système d'exploitation en lui-même. En effet, bon nombre d'étudiants ne connaissaient de MsDos que la commande "turbo", et n'avaient pas la moindre notion de ce qu'est un répertoire. La confrontation avec Unix risquait d'être rude, d'autant plus qu'avec quatre TP de deux heures, le temps nous était compté.

Nous avons choisi de préparer un environnement à base de X, de twm, et d'emacs, cet éditeur étant choisi, entre autres, à cause de ses possibilités d'extensions. Le but était de permettre aux étudiants de ne travailler qu'avec un seul logiciel, leur épargnant ainsi le contact direct avec le système. Nous avons, au départ, visé trois objectifs : l'environnement devait être intuitif, utiliser les touches de fonction, et permettre de passer rapidement de l'édition au test.

Cela nous a amené à brider sciemment emacs pour éviter de fausses manipulations. Le système a beaucoup évolué au cours des TP, au fur et à mesure

que de nouvelles failles se révélèrent. Notre objectif, dans le cadre où nous nous trouvions, était de régler la presque totalité des interactions estudiantines avec Caml et Unix, en empêchant les fausses manoeuvres. Nous n'avons donc pas développé un mode emacs classique, en ce sens que notre mode prend totalement le contrôle de l'éditeur.

Les contraintes étaient de garantir que le nom de tout fichier Caml se terminât par .ml ou mli; que seuls les fichiers Caml fussent exécutables, de fournir une aide disponible facilement (par la touche HELP), et de gérer de façon simple les notions typiquement "emacsiennes" de buffer et de fenêtre, assez difficiles à inculquer à un débutant.

L'un dans l'autre, l'environnement s'est révélé adapté. Les étudiants ont pu se concentrer sur le langage en laissant de côté les notions, intéressantes certes, mais ici encombrantes, liées au système d'exploitation. Ceci dit, les plus curieux (et les plus avancés) ont pu faire d'une pierre deux coups, et s'initier à la fois à Caml et à Unix, sans pour autant que l'on sacrifiât les étudiants plus lents.

V- Conclusion

Le langage Caml apporte une grande souplesse de programmation en partie grâce au système de typage. La simple lecture du résultat de l'exécution de l'algorithme de typage est une première sécurité pour l'étudiant quant au résultat attendu.

Le typage dynamique dans un premier temps de Caml, apparaît pour le débutant comme quelque chose de fondamental et non pas pour un programmeur confirmé, comme un artifice. D'autre part, le typage statique du langage, dans une seconde approche, permet de définir clairement et simplement l'implémentation de paquetage. L'écriture de composants logiciels réutilisables est beaucoup plus simple en Caml qu'avec Ada [Acc92], il n'est pas nécessaire de préciser le nom des différents arguments.

La présentation du cours sous l'angle des styles de programmation selon Harold Abelson [Abe89] a permis une approche beaucoup plus riche et plus pragmatique pour l'étudiant. Les procédures d'ordre supérieur, les notions d'état local et d'objet ont ensuite ouvert de nouvelles possibilités peu explorées encore dans la culture informatique très classique de nos étudiants.

Nous avons d'ailleurs voulu que l'apport de ce nouveau style soit découvert naturellement en interdisant a priori les réflexes du style impératif. L'introduction des mécanismes de la programmation impérative présents au niveau du langage n'a nullement perturbé les étudiants, bien au contraire, la quasi-totalité a continué à programmer dans un style purement fonctionnel.

En conclusion, nous dirons que, malgré le nombre insuffisant d'heures, les étudiants ayant une culture mathématique ont apprécié ce langage, car le code des fonctions en Caml (purement fonctionnel) abordées en exercices était proche de

leurs spécifications mathématiques. L'aspect fonctionnel et interactif a permis aux étudiants de négliger les instructions d'entrées sorties, ce qui n'est pas le cas dans l'initiation en Pascal.

Quelques notions étaient semble-t-il difficiles à appréhender, comme par exemple la récursivité terminale, le typage des fonctions contenant des variables polymorphes.

Mis à part les problèmes cités, nous pensons que la mise en oeuvre par Caml de la modularité, du polymorphisme, de la récupération des erreurs et de la facilité du codage, font que ce langage est un outil pédagogique d'avenir pour l'enseignement des concepts modernes en programmation.

Enfin, en tenant compte des contraintes horaires, on pourra dire que la quasi-totalité des objectifs pédagogiques fixés au départ: apprentissage de la programmation fonctionnelle et de notions informatiques telles que la récursivité, les types, la modularité, l'encapsulation, la généricité, ont été atteints. On notera aussi à la lecture des réponses aux questionnaire, le grand intérêt des étudiants pour un tel enseignement.

Daniel BOURGET

Télécom Bretagne, BP 832 Brest cedex - France

VI- Bibliographie

- [Mil83] **R. Milner**, "A proposal for Standard ML", Report CSR15783, University of Edimburgh}, 1983.
- [Abe89], **H. Abelson and G. Sussman and J. Sussman**}, "Structure et interprétation des programmes informatique", InterEditions, 1989.
- [Acc92], **T. Accart Hardin and V. Donzeau-Gouge Viguié**, "Concepts et outils de programmation, le style fonctionnel, le style impératif, avec Caml et Ada", InterEditions, 1992.
- [Bou93], **D. Bourget**}, "Introduction à Caml Light", Département informatique de Télécom Bretagne, 1993.
- [Ler92], **X. Leroy**, "Manuel de référence de Caml_Light", INRIA, 1992, Rapport Technique.
- [Mau92], **M. Mauny**, "Functionnal programming in CAML", INRIA}, 1992, Rapport Technique 129.
- [Ler93], **X. Leroy and P. Weis**, "Manuel de référence du langage Caml", InterEditions}, 1993.
- [Wei93], **P. Weis and X. Leroy**, "Le langage Caml", InterEditions, 1993.

VII- Annexes

Utilisation d'Emacs

Rappelons que notre système est extrêmement ad hoc, et ne fonctionne que si emacs est utilisé sous X11, avec twm comme gestionnaire de fenêtres.

Lorsque l'étudiant se trouve face au système, la première chose qu'il voit est cet écran d'aide :

ENVIRONNEMENT Caml DANS EMACS

touche	fonction
Undo	annule les changements pre'cedents
Open	charge un fichier source Caml dans emacs
Stop	quitte l'editeur
Help	Affiche cette fenetre
F1	transmet le buffet courant a' l'interpre'te Caml
F2	sauve le fichier courant
F3	charge un fichier source Caml dans emacs
F4	change le buffet affiche'
F5	change de fenetre
F6	avoir une seule fenetres
F12	Lance une session Caml (sans charger de fichier)
C-g	arrete la commande en cours
C-k	efface jusqu'a' la fin de la ligne

Pour quitter l'aide, tapez F4 ou Open

]

Il s'est avéré assez utile, quoique les élèves aient une propension naturelle à poser des questions plutôt que de le consulter. Un environnement à base de menus, réalisable dans emacs, serait peut-être préférable.

Un certain nombre de commandes appellent des commentaires :

- [Open] est le début naturel d'une édition. Par défaut, le suffixe ".ml" est rajouté, mais le suffixe ".mli" est aussi accepté;
- [Stop] sortie de l'éditeur, avec sauvegarde optionnelle. On tue systématiquement le programme Caml, afin de ne pas désorienter l'utilisateur ;
- [F1] interprète le buffer courant, si c'est possible ; on évite d'interpréter autre chose que du Caml.

Fichier .emacs

```

*****
;;; Ceci est un fichier .emacs . Il doit se trouver dans votre HOME.
;;;
... *****
;;; Serge Rosmorduc 28/11/91
;;; Daniel Bourget 28/11/91 10/10/92 20/1/95
;;; Quelques cles pour Emacs pour Caml
...
... *****
;;;

;;; Le pave' nume'rique pour les stations SUN

(define-key global-map "[A" 'previous-line)
(define-key global-map "[B" 'next-line)
(define-key global-map "[C" 'forward-char)
(define-key global-map "[D" 'backward-char)
(define-key global-map "[218z" 'recenter)
(define-key global-map "[222z" 'scroll-up)
(define-key global-map "[216z" 'scroll-down)
(define-key global-map "[214z" 'beginning-of-buffer)
(define-key global-map "[220z" 'end-of-buffer)

... *****
;;;

;;; variables intéressantes:
(setq auto-save-default nil)
(put 'eval-expression 'disabled nil)

;;; Caml

(setq auto-mode-alist (cons (purecopy ("\\.ml?$" .caml-mode))
                             auto-mode-alist))

(defun caml-mode ()
  (indented-text-mode))

;;; teste si le buffer n'est pas *AIDE* ou *Caml*
(defun mauvais-buffer ()
  (or (equal (buffer-name) "*Caml*") (equal (buffer-name) "*AIDE*"))))

```

```
(setq text-mode-hook '(lambda () (auto-fill-mode 1)))

(defun caml ()
  "lance camllight"
  (interactive "")
  (require 'shell)
  (switch-to-buffer (make-shell "Caml" "camllight"))
  (interactive)
  )

(defun send-buffer2Caml ()
  "envoyer un buffet a caml; le nom du buffer DOIT se terminer en .ml!"
  (interactive)
  (if (mauvais-buffer)
      nil
      (if (string-match "\\..mli?$" (buffer-name) )
          (let* ((wherearewe buffer-file-name)
                 (buff-name (current-buffer)))
              (progn
                (if (not (get-process "*Caml*"))
                    (progn (caml)
                           (switch-to-buffer buff-name)))
                (send-region "*Caml*" (point-min) (point-max))
                (send-string "*Caml*" "\n")
                (switch-to-buffer-other-window "*Caml*")))
          (let ((nom-buff (read-string "donnez un nom a` votre fichier : ")))
              (let ((vrai-nom (if (string-match "\\..mli?$" nom-buff)
                                 nom-buff
                                 (concat nom-buff ".ml"))))
                  (rename-buffer vrai-nom)
                  (set-visited-file-name vrai-nom)
                  )
              (indented-text-mode)
              (setq buffer-offer-save t)
              (send-buffer2Caml))))))

;;; chargement de texte ml: F3 ou OPEN

(setq nom-de-repertoire ".")

(defun liste-fichiers ()
  (mapcar 'list (directory-files ".") ))

(defun charge-ml (&optional dir)
  "charge un fichier ml "
  (interactive)
```

```

;;; MODIFIE
(let*
  ((nom-fic (expand-file-name (read-file-name
                              "Nom du fichier: "
                              nil
                              "temp"
                              ))))
  (find-file
   (if (string-match "\\..mli?$" nom-fic )
       nom-fic
       (concat nom-fic ".ml")))))

(defun sauve-cml ()
  "sauve un buffer"
  (interactive)
  (if (mauvais-buffer)
      nil
      (if (string-match "\\..mli?$" (buffer-name) )
          (save-buffer)
          (let ((nom-buff
                 (read-string "Tapez le nom de votre fichier : "))
                (let ((vrai-nom (if (string-match "\\..mli?$" nom-buff)
                                    nom-buff
                                    (concat nom-buff ".ml"))))
                    (rename-buffer vrai-nom)
                    (set-visited-file-name vrai-nom)
                    (save-buffer)
                    )))))

(define-key global-map [f3] 'charge-ml)

;;; sauvegarde de texte ml: F2
(define-key global-map [f2] 'sauve-cml)

;;; F1 chargement dans Caml
(define-key global-map [f1] 'send-buffer2Caml)

;;; f4 changement de buffer
(defun change-buffer ()
  "changer le buffer"
  (interactive)
  (progn
   (bury-buffer (current-buffer))
   (switch-to-buffer (other-buffer (current-buffer) 1)))

```

```

(define-key global-map [f4] 'change-buffer)

;;; f5 changement de fenetre
(define-key global-map [f5] 'other-window)

;;; f6 une seule fenetre
(define-key global-map [f6] 'delete-other-windows)

;;; touche Stop: sortie de l'editeur
(defun sortir () "pas de doc" (interactive)
  (if (get-process "Caml")
      (kill-process "Caml"))
  (sleep-for 1)
  (save-buffers-kill-emacs))

(define-key global-map "[23~" 'sortir)

(defun delateux () (interactive) (delete-char 1))
(define-key global-map "\C-H" 'backward-delete-char-untabify)
(define-key global-map "" 'delateux)
(define-key global-map "[28~" 'lister-aide)
(define-key global-map "[26~" 'undo)
(define-key global-map "[32~" 'yank)
(define-key global-map "[29~" 'copy-region-as-kill)
(define-key global-map "[31~" 'charge-ml)
(define-key global-map "[24~" 'caml)
(define-key global-map "[33~" 'isearch-forward)

;;; On supprime \C-z
(global-unset-key "\C-z")

;;; On Lance caml

(defun lister-aide ()
  (interactive)
  (if (get-buffer "*AIDE*")
      (switch-to-buffer "*AIDE*")
      (let ((aide (get-buffer-create "*AIDE*")))
        (switch-to-buffer "*AIDE*")
        (insert-string "\t ENVIRONNEMENT Caml DANS EMACS\n\
touche\t\tfonction\n\
\n\
Undo\t\t annule les changements pre'ce'dents\n\
Open\t\t charge un fichier source Caml dans emacs\n\
Stop\t\t quitte l'editeur\n\

```

```
Help\t Affiche cette fenetre
F1\t transmet le buffet courant a l'interprete caml\n
F2\t sauve le fichier courant\n
F3\t charge un fichier source Caml dans emacs\n
F4\t change le buffet affiche\n
F5\t change de fenetre \n
F6\t avoir une seule fenetres\n
F12\t Lance une session caml (sans charger de fichier)\n
\n
C-g\t arrete la commande en cours\n
C-k\t efface jusqu'a la fin de la ligne\n\n
Pour quitter l'aide, tapez F4 ou Open\n
\n
")
  (setq buffer-read-only t))))

(lister-aide)
(setq inhibit-startup-message t)
```


Apprendre avec alliance

Marie-Martine PAGET- M.M. Rosset-Louerat- A.Derieux

présenté le vendredi 12 avril 1996 à 12h00

Résumé: *ALLIANCE est un environnement technologique et pédagogique développé par une équipe du laboratoire MASI dans le cadre de la Conception Assistée par Ordinateur de circuits intégrés. Ses objectifs sont doubles: l'enseignement et la recherche.*

Conception de circuits intégrés

Pour permettre d'intégrer plusieurs millions de transistors dans un seul circuit de surface de l'ordre du centimètre carré, il est nécessaire de disposer de puissants outils d'aide à la conception. ALLIANCE permet de décrire les fonctions du circuit que l'on souhaite fabriquer, puis, en allant jusqu'à la dernière étape avant la fabrication, de générer automatiquement le dessin de l'emplacement occupé par les transistors et leurs connexions.

Pédagogie

L'apprenant peut suivre la méthode complète nécessaire à la réalisation d'un circuit intégré depuis ses spécifications jusqu'au dessin des masques. Dans cette démarche conceptuelle formalisée où intervient le concept des différentes vues du circuit intégré, les apports d'ALLIANCE sont multiples. Grâce à ses propres outils et à ses nombreuses bibliothèques il assure à l'utilisateur la maîtrise complète de la conception de son circuit, son indépendance par rapport au fondeur. Mais c'est aussi un environnement ouvert sur les autres chaînes de CAO "commerciales".

Recherche

La chaîne ALLIANCE permet le développement de nouveaux concepts et de nouveaux outils avancés, qu'utilisent les industriels. Entre 1993 et 1995 trois gros circuits ont réellement été faits, l'un permettant l'interconnexion de réseaux de communication très rapides, un coprocesseur pour applications biologiques et un microprocesseur performant.

Apprendre avec Alliance

Par la description des outils et des tutoriels associés, notre exposé devrait permettre à la fois de répondre aux questions :

"Qu'apprend on avec ALLIANCE?"

"Comment apprend on avec ALLIANCE?"

La problématique sera posée de son succès auprès de nombreuses universités dans le monde, bien qu'il n'ait pas été conçu comme un véritable produit pédagogique. Dans la mesure où un enseignement à la conception de circuits a

besoin d'une chaîne de CAO, ALLIANCE est complète, plus simple d'utilisation que les autres chaînes, gratuite et fonctionnant sur un PC.

Apprendre avec Alliance

Résumé: Nous présentons la chaîne Alliance, un environnement d'apprentissage dédié à la conception de circuits VLSI. Alliance a deux objectifs : l'un de type pédagogique puisqu'elle est utilisée pour l'enseignement dans le cadre d'une formation des étudiants à la conception de circuits intégrés, l'autre de type prospectif en recherche puisqu'elle sert de base pour la formalisation de nouvelles idées et l'expérimentation de nouveaux outils logiciels de CAO. Elle est déjà très utilisée telle quelle dans le monde universitaire et industriel, mais nous nous interrogeons ici sur ce qu'elle enseigne et comment elle le fait, puisqu'il s'agit là de la mise en place réussie de stratégies particulières d'apprentissage.

1. Introduction

Alliance est une chaîne complète d'outils et de bibliothèques portables destinée à la CAO (Conception Assistée par Ordinateur) de circuits intégrés VLSI (Very Large Scale Integration) [GRE92-1, BAZ94-2]. Alliance est disponible gratuitement sur réseau Internet et peut être installée sur des stations de travail UNIX ou sur des PC 386 sous Linux .

C'est plus qu'un ensemble d'outils et de bibliothèques, c'est aussi une pédagogie adaptée à un apprentissage particulier. En effet pour concevoir de gros circuits intégrés (quelques centaines de milliers de transistors) il faut suivre une méthode qui s'apparente à la programmation structurée d'un gros logiciel. C'est un travail d'électronique et d'informatique à la fois, qui a pour but de produire le dessin optimisé des masques nécessaires pour la fabrication du circuit. Le procédé technologique de fabrication consiste à utiliser les masques dessinés par le concepteur pour graver dans le silicium plusieurs niveaux (couches) selon une séquence bien définie.

2. Méthodologie de conception d'un circuit intégré

Nous allons d'abord décrire brièvement les étapes et la méthodologie de conception d'un circuit intégré. Ceci devrait nous permettre déjà de répondre à la question: "Qu'apprend-t-on avec Alliance ?".

Un circuit intégré possède trois types de description (Figure 1) : une vue comportementale qui décrit la fonction qu'il doit réaliser, une description structurelle hiérarchique (ou netlist) qui représente le schéma en transistors et une vue physique ou layout qui fait apparaître le dessin des masques de fabrication. La vue comportementale donne les sorties en fonction des entrées. La vue structurelle utilise des schémas de base dont on dispose dans des bibliothèques.

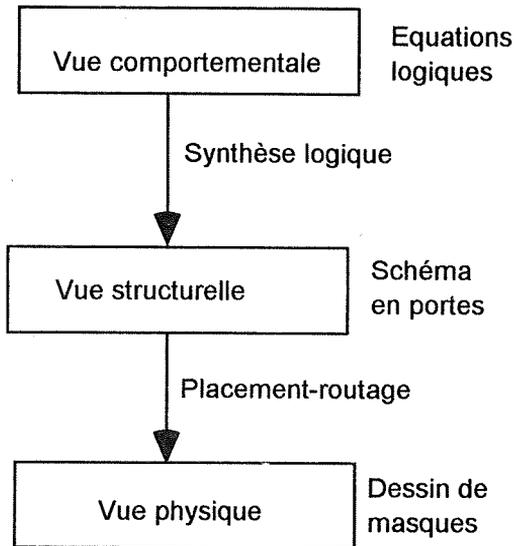


Figure 1 : Concept des vues d'un circuit intégré

Pour passer automatiquement de la vue comportementale en langage VHDL (Very High Speed Integrated Circuit Hardware Description Language) à la vue structurelle, on fait une synthèse logique. Entre le schéma en portes et la vue physique une opération de placement routage est nécessaire. Des outils permettent les opérations inverses, un extracteur de netlist de la vue physique vers la vue structurelle, un abstracteur fonctionnel pour établir le comportement à partir d'une netlist.

Alliance est une chaîne ouverte. Chaque outil peut fonctionner indépendamment des autres. Les formats de fichiers utilisés sont des formats standard de l'industrie (VHDL, EDIF, SPICE, GDSII). Il y a interopérabilité avec d'autres chaînes de conception du commerce comme par exemple la chaîne Compass en ce qui concerne les représentations layout et netlist. Une structure de données internes commune, de type orienté objet permet de représenter les trois vues [PET94].

Signalons une conséquence du succès d'Alliance et de sa capacité à supporter des formats différents des siens : l'entreprise BULL a décidé d'interfacer son propre logiciel de routage cheops avec Alliance. Ce logiciel fait actuellement partie de la version 3.0 de la chaîne Alliance.

La méthodologie de conception est un ensemble séquencé d'opérations utilisant une bibliothèque de cellules prédéfinies dans une approche hiérarchique et structurée, où un objet sera décrit par un nombre limité d'objets standardisés plus simples. Pour réaliser un circuit "zéro défaut" on utilise une conception descendante, et une validation ascendante.

3. Les outils d'Alliance

Les outils d'Alliance qui représentent plus de 500.000 lignes de langage C sont le fruit d'une réalisation collective, qui a d'ailleurs donné lieu au prix spécial d'équipe du concours Seymour Cray en 1994.

Le découpage modulaire permet une intégration plus facile des nouveaux outils par étapes successives. La Figure 2 qui complète la Figure 1, illustre les interactions entre les outils et les structures de données et fait apparaître les étapes essentielles de la méthode de conception et de validation d'un circuit avec Alliance.

Un circuit est représenté par un cœur entouré par une couronne de plots. Les plots constituent l'interface entre le cœur qui réalise la fonction du circuit et l'extérieur du circuit. Les plots sont reliés aux broches du boîtier dans lequel sera encapsulé le circuit. Le cœur d'un processeur peut être décomposé en une ou plusieurs parties opératives ou chemins de données, et une ou plusieurs parties contrôles. Ces éléments ont des caractéristiques propres et nécessitent des outils de conception différents.

4. Les tutoriels d'Alliance

Une fois décrite la chaîne Alliance, ses outils et la méthode qui a été adoptée pour mener à terme la conception d'un circuit depuis sa spécification jusqu'aux masques, nous allons présenter les aides qui lui sont associées afin de préciser "Comment on apprend avec Alliance".

Les choix pédagogiques pris pour cet environnement le placent selon Mendelsohn [MEN92] à la fois parmi les logiciels ouverts et parmi les systèmes spécialisés construits pour l'acquisition d'une expertise bien définie.

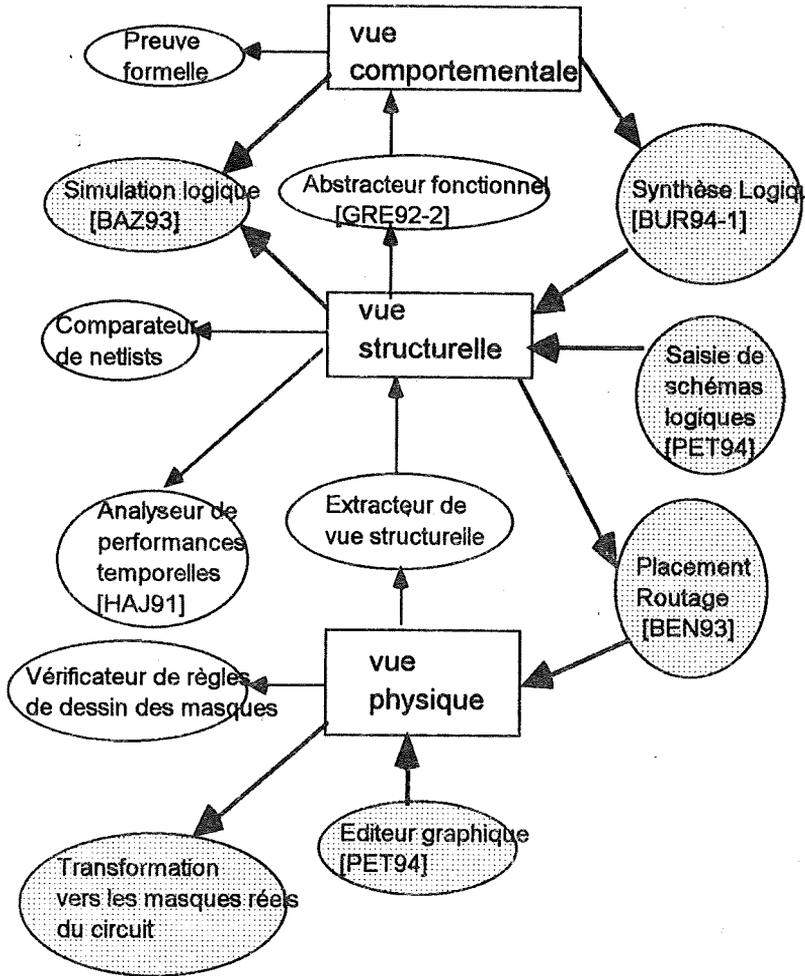
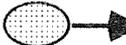


Figure 2 : Formalisation de la démarche conceptuelle

-  : outils de conception (top-down)
-  : outils de validation (bottom-up)

Les aides sont des tutoriels car ce sont des séquences d'apprentissage au sens le plus complet du terme, qui sont mises à la disposition de l'apprenant

comme un tuteur, surtout quand il y a une part d'autoformation [DEM92]. Ces tutoriels, en langue anglaise, se classent en deux catégories, ceux qui concernent la méthodologie et ceux qui s'appliquent à un outil ou bien à une fonction du cycle complet.

4.1 Tutoriels concernant la méthodologie

4.1.1 Rôle des tutoriels concernant la méthodologie

Le but de ces tutoriels est d'apprendre la méthodologie de conception et de validation d'un circuit intégré avec Alliance. Trois circuits de complexités différentes ont été choisis. Les tutoriels correspondants peuvent être utilisés pour prendre en main les outils de la chaîne et suivre la réalisation du circuit en partant de sa spécification fonctionnelle.

Dans chacun des cas tous les fichiers sources sont fournis avec le tutoriel. Un ordre de grandeur du temps nécessaire pour la construction du circuit est indiqué (deux heures pour l'amd2901 et six heures pour le dlx sur une station Sparc). Ces temps représentent le temps passé à exécuter le tutoriel et non le temps de réflexion et de compréhension.

Il est possible de faire la construction ou bien pas à pas en tapant les commandes successives indiquées ou bien automatiquement en exécutant le fichier Makefile fourni, qui gère l'ensemble de la construction du circuit considéré. Tels qu'ils existent, ces trois tutoriels ne sont pas seulement trois exemples différents mais trois exemples de plus en plus difficiles à aborder, mettant en œuvre de plus en plus d'outils d'Alliance, pour permettre à l'apprenant une familiarisation de plus en plus approfondie avec la chaîne .

Au cours des opérations, la seule interactivité possible de l'apprenant est localisée dans le choix d'une option plutôt qu'une autre au niveau des commandes.

4.1.2 Tutoriel 1 : Additionneur 4 bits

Le circuit addacu, qui devrait être étudié en premier avec Alliance, est très simple (environ 500 transistors); il est composé d'un additionneur 4 bits, d'un registre 4 bits et d'un multiplexeur. Ce tutoriel fait partie d'Alliance depuis sa première version (Alliance 1.0, septembre 1992).

4.1.3 Tutoriel 2 : AMD2901 4 bits

Le second circuit est déjà plus complexe (environ 3 000 transistors), c'est le processeur 4 bits amd2901 [AMD77] . La conception est faite par saisie textuelle des schémas en portes logiques. Ce tutoriel fait partie d'Alliance depuis sa première version (Alliance 1.0, septembre 1992).

4.1.4 Tutoriel 3 : Microprocesseur 32 bits DLX

Quant au troisième circuit, c'est le microprocesseur 32 bits dlx [HEN92] d'environ 30 000 transistors. La conception est faite en tenant compte du découpage du microprocesseur en un bloc de contrôle et un chemin de données (partie

opérative). Le contrôle est décrit sous forme d'automate. Les outils d'Alliance mis en œuvre sont les mêmes que pour le Tutoriel 2, avec en plus les outils de synthèse. Pour faciliter la simulation et le développement de nouveaux vecteurs de simulation, un environnement particulier a été créé. Il s'agit d'une carte, décrite en VHDL comportant le microprocesseur *dlx*, un décodeur d'adresse et des mémoires ram et rom. Un assembleur est utilisé pour transformer les instructions du *dlx* en fichier interprétable par le simulateur logique d'Alliance. Ce tutoriel fait partie d'Alliance depuis sa deuxième version (Alliance 2.0, février 1993).

Une partie du Tutoriel 3 décrivant la synthèse de l'automate par utilisation de l'outil *syf*, est donnée ci-dessous :

3.2 Structural Design

3.2.1 Design Hierachy

In this step, the structural descriptions of the chip (*dlxm.vst*), and the core (*core.vst*) are used. The chip is described as a core surrounded by pads. The core is divided into two structural blocks: control and data path, the control block being also divided into two structural blocks : sequencer and status, each of which must be represented by its own behavioural description. The following source files are provided :

- *dlxm_chip.vst* : VHDL structural model of the *dlxm* chip instantiating core and pads.
- *dlxm_core.vst* : VHDL structural model of the core instantiating the data path, and the control.
- *dlxm_ctl.vst.h* : VHDL structural model of the control instantiating the sequencer and the status (the control model *dlxm_ctl.vst* being saved in the file *dlxm_ctl.vst.h* to prevent future erasing).
- *dlxm_seq.fsm* : VHDL finite state machine model of the sequencer.
- *dlxm_sts.vbe* : VHDL model of the status.
- *dlxm_dpt.vbe* : VHDL model of the data path.

3.2.2 Sequencer state assignment

The sequencer is written using a subset of VHDL specifically designed for the description of finite state machines. You must therefore compile this into a VHDL data-flow behavioural model (*vbe*) using the ALLIANCE tool *syf* :

```
>syf -s dlxm_seq -of dlxm_seq -scan -save
```

- *-s* : uses a vertical encoding algorithm
- *dlxm_seq* : fsm source file (*dlxm_seq.fsm*)
- *-of dlxm_seq* : output behavioural description (*dlxm_seq.vbe*)
- *-scan* : adds a scan path to the state register
- *-save* : saves encoding result in *dlxm_seq.cod* file

Ce tutoriel a l'avantage d'illustrer la méthodologie d'Alliance sur un circuit complexe et pourtant accessible et "connu" puisque le processeur dlx a été créé par Hennessy et Patterson comme un modèle typique de processeur RISC pour l'enseignement. Il est cependant manifeste que cet exemple est trop complexe pour qu'un utilisateur d'Alliance qui ne connaîtrait pas au préalable le dlx puisse comprendre, en exécutant le tutoriel, le contenu des fichiers sources, en particulier les fichiers décrivant l'automate de contrôle et le chemin de données.

4.2 Tutoriels concernant les outils

L'utilisateur dispose à la fois de la documentation en ligne et d'aides spécifiques concernant une fonction particulière du cycle de conception.

4.2.1 Documentation en ligne

Depuis la création d'Alliance il existe une documentation en ligne pour chaque outil de la chaîne Alliance, utilisant la commande `man` du système UNIX. En entrant la commande `man nom_de_l'outil` on obtient les informations concernant l'outil correspondant d'Alliance. Actuellement plus de mille pages de documentation de ce type sont consultables par le développeur. Un exemple tel qu'il est disponible, c'est à dire en anglais, est donné en annexe pour l'outil `syf` de synthèse d'automate. Ce type de documentation n'est que le mode d'emploi de l'outil.

4.2.2 Tutoriels spécifiques

Dans la première version d'Alliance, seuls existaient les tutoriels 1 et 2 présentés précédemment. L'arrivée des outils de synthèse logique et des outils de conception orientés chemin de données a fait apparaître la nécessité d'un nouveau tutoriel illustrant la méthodologie accompagnant ces nouveaux outils. Ainsi fut créé le Tutoriel 3. Le circuit qu'il présente est un microprocesseur 32 bits microprogrammé. A l'usage, ce tutoriel s'est révélé inefficace, car trop complexe pour enseigner la méthode de conception d'un circuit. Il est apparu qu'il fallait utiliser des circuits de fonctionnalité plus simple pour mettre l'accent sur une certaine étape de la conception.

C'est ainsi que de nouveaux tutoriels ont été conçus pour la dernière version (Alliance 3.0, mai 1995) qui présentent séparément les outils de conception, tout en permettant la conception totale du circuit et sa validation. Sont ainsi disponibles :

- un tutoriel décrivant la conception d'un circuit digicode à partir de sa spécification sous forme d'une machine à états finis, pour insister sur les outils de synthèse logique [BUR94-1].

- un tutoriel décrivant la conception de la partie opérative d'un circuit à partir de sa spécification structurelle, pour insister sur les outils de conception d'un chemin de données optimisé [BEN93].

- un tutoriel décrivant la réalisation du circuit digicode sur FPGA (Field Programmable Gate Array) par l'outil Alligator (ALLiance GATE Array OptimizeR). Alligator [BUR94-2] prend en entrée une vue comportementale VHDL et fournit une netlist de portes disponibles sur matrice prédéfinie, au format

particulier du fabricant. L'intérêt des FPGA est de permettre la réalisation rapide et à faible coût d'un prototype du circuit puisque, une fois la matrice disponible, seuls les connexions entre portes sont à réaliser.

Malheureusement il n'existe pas un tutoriel pour chaque outil et il reste du travail à faire pour que les tutoriels permettent réellement un apprentissage autonome.

En fait la chaîne est surtout utilisée pour des étudiants de maîtrise ou DEA, qui ne sont pas livrés à eux mêmes et auxquels des enseignants donnent les explications nécessaires. Si l'on souhaite voir Alliance utilisé par des apprenants moins avertis, il faudrait qu'ils disposent d'exercices pratiques supplémentaires avec questions et réponses pour favoriser leur apprentissage. Chaque tutoriel serait composé d'une série d'unités comprenant chacune une présentation d'informations et de commandes à faire exécuter, accompagnée de sollicitation de l'apprenant, saisie de sa réponse et traitement du message correspondant avec réaction à ce message pour contrôler le bon acquis des notions à faire passer. En plus de compétences spécifiques sur l'expertise du contenu, ceci suppose des compétences didactiques pour déterminer les activités à proposer aux apprenants, types d'exercices et de commentaires, avec des niveaux différents allant du très débutant au débutant avancé. Ainsi faire d'Alliance un environnement de simulation pédagogique plus structuré et beaucoup plus interactif et convivial, reste une amélioration délicate, qui n'est pas du ressort des chercheurs en CAO et VLSI.

Dans les tutoriels un effort de présentation des écrans et l'utilisation de plusieurs fenêtres de visualisation apporteraient de la convivialité et éviteraient une lecture passive d'écrans successifs. On sait bien que l'ergonomie joue un rôle important dans la relation apprenant-machine. Celle ci a par exemple été particulièrement soignée dans [EDEC95], un ensemble de cours interactifs concernant la conception à tous les niveaux de circuits électroniques, mis en place dans le cadre de la collaboration de plusieurs universités du Royaume Uni.

Une synthèse générale de l'ensemble de la chaîne Alliance permettrait aussi de se repérer au milieu des nombreux outils, sans avoir recours à un exposé préalable. Ceci s'est d'ailleurs déjà fait sentir puisqu'il existe quelques documents comme par exemple celui qui précise les structures pour l'outil de simulation logique Asimut [P. BAZARGAN SABET et H.N VUONG "Les nouvelles structures pour Asimut", Mai 1995], qui n'est pas un tutoriel mais un document d'information sur un point particulier.

Malgré ces aides plus ou moins tutorielles, il reste aux apprenants des difficultés à surmonter. Par exemple, l'un des utilisateurs a écrit (traduit de l'anglais) "J'ai essayé d'apprendre Alliance en exécutant les trois tutoriaux. Après avoir utilisé les exemples de addacu et amd2901 il ne me semble pas très clair d'écrire les fichiers C nécessaires à l'outil de saisie textuelle de schémas Genlib. Les informations des man correspondants n'aident pas assez à comprendre comment

manipuler l'écriture du code". Une trop grande liberté d'action rend souvent l'apprentissage difficile.

4.3 *Expérience d'enseignement à l'Université Pierre et Marie Curie*

Alliance est un environnement évolutif paramétrable par les enseignants. Il sert de base autour de laquelle se construit un enseignement dispensé aux étudiants. Il est utilisé à l'Université Pierre et Marie Curie en maîtrise d'informatique, en maîtrise EEA (Electronique, Electrotechnique et Automatique), au DEA ASIME (Architecture des Systèmes Intégrés et Micro Electronique) et au DESS CIMI (Circuits Intégrés et Micro Informatique). Depuis plusieurs années, nous avons mis au point une formation de deux mois (octobre, novembre) permettant aux étudiants provenant de cursus différents (Informatique, Electronique, Physique), l'apprentissage de la chaîne de CAO Alliance. Cette formation a été conçue pour les étudiants désirant suivre le DEA ASIME ou le DESS CIMI. Chaque année se joignent à ces étudiants des auditeurs libres, qui sont en général des professeurs étrangers (deux à trois par an, en provenance du Brésil, d'Espagne et d'Egypte) désirant suivre cette formation afin d'utiliser la chaîne de CAO Alliance lors de leur enseignement de la microélectronique.

Des cours magistraux sont dispensés le matin du premier mois, sur les thèmes suivants:

- Méthodologie de conception d'un circuit
- Présentation des outils de CAO
- Circuiterie CMOS
- Architecture du processeur DLX.

Les après-midis sont consacrés à des travaux pratiques d'utilisation de la chaîne Alliance. Des exercices sont proposés aux étudiants qui travaillent en binôme:

- La première semaine, le travail consiste en un apprentissage du langage VHDL: la représentation comportementale et structurelle d'un additionneur accumulateur.

- La deuxième semaine, le travail demandé concerne les outils de synthèse logique et le routage de cellules standard, ainsi que le dessin des masques d'une cellule de la bibliothèque. L'exemple de circuit utilisé est un digicode. Les étudiants ont à décrire l'automate de ce digicode sous la forme d'un automate d'états finis qui est ensuite synthétisé en cellules standard, puis routé, et validé selon la méthodologie explicitée en cours (Figure 2).

- Durant la troisième semaine, on réalise l'AMD2901 en deux blocs distincts: la partie contrôle est conçue en cellules standard et la partie opérative est obtenue à l'aide du compilateur de chemin de données.

- La quatrième semaine est consacrée aux problèmes de simulation électrique et de vérification temporelle. La simulation électrique est obtenue par utilisation du logiciel SPICE [HMU90] tandis que l'analyse temporelle peut être obtenue à l'aide de l'outil d'Alliance TAS [HAJ91], ce qui permet d'aborder les notions de stabilité et en particulier l'étude des contraintes de type "temps maximal

d'établissement d'un signal" et de "temps minimal de stabilité d'un signal" à respecter pour le bon fonctionnement d'un circuit avec points mémorisants. On met en évidence les différences existantes entre un simulateur électrique (précision, nécessité de stimuli, mais lenteur) et un analyseur temporel (moins précis, plus rapide et sans nécessité de stimuli).

Cette formation allie le contenu des tutoriels précédemment décrits à des explications sur les concepts nécessaires à la conception de circuits VLSI digitaux (Figure 1 et Figure 2).

Après ce mois de formation, nous considérons que les étudiants sont alors capables de réaliser un projet important le microprocesseur 32 bits DLX, ainsi le mois de novembre est entièrement consacré à la réalisation de ce circuit en respectant la méthodologie d'Alliance. Pour ce faire, par groupes de cinq à six personnes, les étudiants partent d'une description fonctionnelle qui constitue la spécification du processeur et réalisent le dessin des masques du circuit. Ce projet permet de faire évoluer le Tutoriel 3 DLX [BAZ94-1] d'une année à l'autre.

5. Autres objectifs

Au delà de l'acquisition des savoir-faire liés à la description d'actions à effectuer pour atteindre un but, la chaîne Alliance a aussi été utilisée pour la conception de circuits de très haute complexité. Trois circuits d'une complexité de plusieurs centaines de milliers de transistors ont été conçus avec Alliance : il s'agit d'un routeur de message utilisant des liens séries à 1 giga-bit/s [ZER95], d'un coprocesseur à architecture SIMD pour le traitement du génome [ARC94] et d'un microprocesseur superscalaire à mot d'instruction très large [GRE94].

Certains outils d'Alliance sont particulièrement originaux et appréciés des industriels, qui n'ont pas forcément à leur disposition un outil industriel équivalent plus performant. C'est le cas :

- du compilateur de chemin de données [BEN93]
- des outils de synthèse logique [BUR94-1], [BUR94-2]
- des outils de vérification TAS [HAJ91], DESB [GRE92-2].

Nous ne pouvons citer ici qu'une partie des publications concernant la chaîne, elles sont en nombre déjà supérieur à 40...

6. Les utilisateurs d'Alliance hors l'Université P. et M. Curie

6.1 Diversité des utilisateurs d'Alliance

Alliance n'est pas seulement utilisée à Paris ! Comme le montre le Tableau 1, Alliance est utilisée dans plus de 150 universités françaises et étrangères dans le monde. Pour cet objectif d'apprentissage, elle peut être installée sur des machines peu coûteuses de type PC sous Linux.

Allemagne	16	Etats Unis	48	Portugal	2
Arabie Saoudite	1	France	18	Roumanie	2
Australie	3	Grèce	1	Royaume Uni	9
Belgique	1	Hong Kong	1	Russie	2
Brésil	6	Inde	1	Singapour	1
Bulgarie	1	Italie	6	Slovenie	3
Canada	2	Iran	2	Suède	3
Colombie	1	Israel	2	Suisse	2
Corée	1	Japon	7		
Danemark	1	Mexique	1		
Egypte	1	Norvège	2		
Espagne	3	Pays Bas	2		
Estonie	1	Pologne	4	TOTAL	157

Tableau 1 : Sites utilisateurs d'Alliance (Janvier 1995)

De plus la diffusion gratuite des fichiers sources sur le réseau Internet par ftp depuis 1992 donne la possibilité aux utilisateurs d'apporter leur réelle participation pour modifier ou corriger chaque nouvel outil. Une vingtaine de messages arrivent chaque semaine, la grande majorité pour demander des compléments d'information, et dont certains ont déjà permis la correction de plusieurs bugs.

6.2 Expériences d'utilisation

A la demande de deux instituts européens, des enseignants du Laboratoire MASI ont été enseigner la méthodologie de conception de circuits intégrés VLSI avec Alliance en Italie et en Pologne.

En novembre 1994, N. Abdallah a fait des cours à l'Université des Nations Unies dans le cadre du "Third Course on Basic VLSI Design Technics". On peut signaler que les organisateurs du cours avaient réalisé pour cette occasion un document composé de sept exercices de difficulté croissante à faire exécuter l'un après l'autre par l'apprenant [ICTP94]. Ces exercices ressemblent beaucoup par leur structure aux différents tutoriels d'Alliance, leur différence porte sur les circuits qui sont étudiés. La réussite de ce cours, proposé à des apprenants de pays très variés, a fait que les organisateurs ont demandé à renouveler l'expérience en 1995.

En mai 1995, E. H. Rejouan est allé installer et enseigner Alliance à l'Université de Gliwice en Pologne où Alliance a été choisie comme logiciel d'enseignement de la conception VLSI. Une coopération entre notre université, l'Université de Silésie et l'Université de Tübingen fait d'ailleurs l'objet d'un projet Européen pour l'enseignement TEMPUS.

Conclusion

D'une façon analogue à la définition donnée pour le Génie logiciel, la chaîne Alliance est un ensemble d'outils et de procédures relatives aux différentes phases de l'évolution d'un circuit intégré depuis sa spécification jusqu'à sa production avec les tests, la documentation, etc... Elle comprend une trentaine d'outils et elle est bien particulière, parce que d'une part elle a le but pédagogique concret de servir pour l'enseignement de toutes les étapes de conception d'un circuit intégré et d'autre part elle comprend des outils avancés utilisés par les industriels. La méthode et les outils proposés sont donc "à l'état de l'art" du point de vue industriel, tout en permettant aux universitaires de montrer le cycle complet de conception ou bien une étape du cycle, qu'ils ont choisi de détailler.

Alliance permet donc à la fois l'acquisition des concepts méthodologiques conduisant à la conception des circuits intégrés et le développement de nouveaux concepts. C'est le type même d'environnement technologique et pédagogique approprié à un milieu spécifique, celui des concepteurs de circuits VLSI. Les difficultés rencontrées par les apprenants sont bien évidemment liées au milieu concerné, celui des développeurs.

En conclusion il ressort qu'Alliance n'est pas du tout conçue comme un véritable produit CBL (Computer Based Learning) analogue au cours d'électronique du Consortium d'Universités anglosaxonnes EDEC [EDEC95] et que pourtant il est très utilisé (150 universités dans le monde), peut-être comme le dit PONTA [PON94] parce que : "Students like much more working with a kind of tool instead of following a straight CBL package. Playing with a tool yields more than reading a text".

S'il est vrai que dans sa version actuelle, utiliser Alliance nécessite une formation préalable à la conception de circuit, il faut aussi souligner qu'un tel enseignement ne peut pas se concevoir sans l'utilisation conjointe d'une chaîne de CAO. Or ces chaînes "commerciales" sont chères, complexes d'utilisation et souvent liées à un fabricant de circuit particulier. Elles s'adressent toujours à un public déjà compétent en architecture de circuits. Dans ce contexte, l'intérêt d'Alliance est de rendre à tous complètement accessibles (sources et structures de données) des outils professionnels de conception VLSI ainsi que la méthode pour les utiliser.

Remerciements

Les auteurs remercient TOUS les membres de l'équipe CAO-VLSI du laboratoire MASI (Méthodologies et Architectures des Systèmes Informatiques), qui ont développé la chaîne Alliance et qui l'ont testée en concevant des circuits complexes.

Marie-Martine PAGET

Maître de conférences, Université Pierre et Marie Curie
Laboratoire Masi, 4 place Jussieu 75252 Paris Cedex

Bibliographie

- [AMD77] "Data sheet of the amd2901", 1977
- [HMU90] L.V. HMURCIK, M. HETTINGER, K.S. GOTTSCHALCK, F.C. FICHTEN : "SPICE Applications to an Undergraduate Electronics Program, IEEE Trans. Educ., 1990, vol. 33, n.2, pp 183
- [HAJ91] A. HAJJAR, R. MARBOT, A. GREINER, P. KIANI : "TAS. An Accurate Timing Analyser for CMOS VLSI", The European Conference on Design Automation, Feb 1991, pp 261-265
- [DEM92] F. DEMAIZIERE, C. DUBUISSON : "De L'EAO aux NTF. Utiliser l'ordinateur pour la formation", Ophrys 1992
- [MEN92] P. MENDELSON : "L'ordinateur dans l'enseignement", Actes de la troisième rencontre francophone de didactique de l'informatique, Sion, Suisse, Juillet 1992, pp 53-62
- [GRE92-1] A. GREINER, F. PECHEUX : "Alliance. A complete Set of CAD Tools for Teaching Digital VLSI Design", Proc Third Eurochip Workshop on VLSI Design Training, Grenoble, Septembre 1992, pp 230-237
- [GRE92-2] A. GREINER, M. LAURENTIN, R. MARBOT : "DESB. A fonctionnal Abstractor for CMOS VLSI circuits", EURO-DAC 92, European Design Automatisisation Conference, Hamburg, Germany, Septembre 1992, pp 22-27
- [HEN92] J. L. HENNESSY, D. A. PATTERSON : "Architecture des ordinateurs - Une approche quantitative", McGraw Hill pour l'édition française, Paris, 1992
- [BAZ93] P. BAZARGAN-SABET, A. GREINER, H N. VUONG : "Asimut A Public Domain VHDL Simulation Tool", Proc. Fourth Eurochip Workshop on VLSI Design Training, Toledo, Spain, Septembre 1993, pp 62-65
- [BEN93] L. BEN AMMAR, A. GREINER : "FITPATH A Process-Independent Datapath Compiler Providing High Density Layout", IFIP Transactions, Synthesis for Control dominated Circuits, G. SAUCIER and J. TRILHE Editors, IFIP North-Holland, 1993, pp 133-151
- [ARC94] D. ARCHAMBAUD, P. FAUDEMAY, A. GREINER : "Rapid-2 an Object-Oriented Associative Memory Applicable to Genome Data Processing", 27th Hawaii International Conference on System Sciences, Hawaii, Janvier 1994, vol.5, pp 150-159
- [GRE94] A. GREINER, L. LUCAS, F. WAJSBURT, L. WINCKEL : "Design of a high complexity superscalar microprocessor with the portable IDPS Asic Library", The European Conference on Design Automation, EDAC EURO-ASIC ETC, Mars 1994, pp 9-13
- [PET94] F. PETROT : "Outils d'Aide au Développement de Bibliothèques VLSI Portables", Thèse de l'Université Pierre et Marie Curie, Paris 6, Juillet 1994
- [PON94] D. PONTA, G. DONZELLINI : "Learning Electronics with Hypermedia and Computer Tools", International Conference on Computer Aided Learning and Instruction in Science and Engineering, CALISCE'94, Telecom Paris, France, Aout 1994, pp 7-23

- [BUR94-1] L. BURGUN, N. DICTUS, A. GREINER, E. PRADO LOPES, C. SARWARY : "Multilevel Logic Optimization of very high complexity circuits", EURO-DAC, Grenoble, Septembre 1994, pp 14-19
- [BUR94-2] L. BURGUN, A. GREINER, E. PRADO LOPES : "A Consistent Approach in Logic Synthesis for FPGA architectures", ASICON, Pekin, Octobre 1994, pp 104-107
- [BAZ94-1] P. BAZARGAN-SABET, J. DUNOYER, A. GREINER, M.M. ROSSET LOUERAT : " A tutorial for Advanced VLSI Course Designing the 32 bit DLX Microprocessor with the Alliance CAD system", Proc Fifth Eurochip Workshop on VLSI Design Training, Dresden, Germany, Octobre 1994, pp 120-127
- [BAZ94-2] P. BAZARGAN SABET, L. BURGUN, A. GREINER, F. PETROT : "Methodology of Development of a Complete CAD System for Digital VLSI Design", VIII Simposio Brasileiro de Concepcao de Circuitos Integrados, Gramado, Bresil, Novembre 1994, pp 11-15
- [ICTP94] Third Course on basic VLSI Design Techniques, International Atomic Energy Agency, United Nations Educational Scientific and Cultural Organization, International Centre for Theoretical Physics, Microprocessor Laboratory, Trieste, Italie, Novembre et Décembre 1994
- [ZER95] B. ZERROUK, V. REIBALDI, F. POTTER, A. GREINER, A. DERIEUX: "A High Speed Message Routing Device Prototype", Rapport Interne de l'Institut Blaise Pascal MASI 95/05, 4 Place Jussieu, 75252 Paris Cedex 05.
- [EDEC95] Electronic Design Education Consortium, "Computer based courseware for the interactive learning of electronic design", ED&TC95, University booth demonstrations, Paris, France, Mars 1995

Annexe : Documentation en ligne fournie pour l'outil de synthèse d'automate syf.

SYF(1)	CAO-VLSI Reference Manual	SYF(1)
NAME	SYF	
	- Finite State Machine synthesis.	

ORIGIN

This software belongs to the ALLIANCE CAD system from the CAO-VLSI team at MASI laboratory, University P. and M. Curie. 4, place Jussieu ; 75252 PARIS Cedex 05 FRANCE FAX: (33-1) 44.27.62.86 ; E-mail cao-vlsi@masi.ibp.fr

SYNOPSIS

syf [-s|-m|-a|-h|-r] FILE [-of output_file_name] [-stack] [-pseudorom] [-scan] [-save]

DESCRIPTION

syf is a Finite State Machine synthesizer. syf allows a fast generation of VHDL Data_Flow description (for further information about this VHDL subset, read the "vbe" manual) from a VHDL Finite State Machine description (see fsm(5)). The input FSM specification can use an internal STACK. Both MOORE and

MEALEY FSM can be synthesized. For a MOORE FSM, a timing-optimized implementation that emulates a ROM with microsequencer is possible. A scan-path for the state registers can be also implemented.

ENVIRONMENT VARIABLES

MBK_WORK_LIB gives the work directory path.

OPTIONS

- s Uses a "vertical" encoding algorithm.
- m Uses a "horizontal" encoding algorithm.
- a Uses a random encoding.
- h Uses an encoding given by user through <file>.cod file. In this file, all line started by a # character is not interpreted. A valid line contains one state name followed by his hexadecimal code.
- r Uses simulated annealing as state assignment algorithm.
- scan With this option syf implements a scan-path for the state registers. Refer to fsm(5) for information about scan-path description.

-of output_file_name

With this option the output file generated by SYF will be out_put_file_name. Else the output file name will be the input file name added to the encoding option. If the -save option is used the encoding result is saved in <input_file_name.cod>.

-stack This option is only useful for STACK FSM. With this option SYF optimizes the state encoding in order to minimize the stack area. This option being time-consuming is optional.

-pseudorom

This option is only available for MOORE FSM. With this option, syf emulates a ROM with micro-sequencer implementation : there is no combinatorial logic between the state registers and the FSM outputs, can be mandatory for external timing constraints.

-save With this option syf saves the encoding result in the <file(encoding algorithm identifier)>.cod. This file has the same syntax as <file>.cod file which is used by

-h option.

EXAMPLE

Environment variables: setenv MBK_WORK_LIB /users/Michel/fsm

syf is called as follow (the dlx_ctrl.fsm is already created in /users/Michel/fsm):

syf -s dlx_ctrl -save

Two files will be generated, a states encoding file dlx_ctrls.cod and a VHDL data flow file /users/Michel/beh/dlx_ctrls.vbe

SEE ALSO vhdl.

DIAGNOSTICS

Please, report problems or bugs to cao-vlsi@masi.ibp.fr

Thème6 : L'Enseignement Professionnel

L'enseignement de l'informatique-outil dans l'enseignement technique supérieur

Robert WALD

présenté le jeudi 11 avril 1996 à 15h45

Thème proposé :

L'enseignement de l'informatique-outil dans l'enseignement technique supérieur.

Ce thème tentera d'approfondir les réflexions conduites lors des Rencontres Francophones de Sion (Suisse) en 1992 et de l'Université d'été de Namur (Belgique) en 1994.

Problématique :

L'apprentissage de certains outils informatiques évolués -notamment les tableurs des dernières générations- permet-il de mettre en oeuvre à travers une pédagogie de projet, un enseignement original de certains concepts fondamentaux de l'informatique tels :

- la méthodologie
- l'analyse des données
- l'algorithmique
- la programmation et la manipulation d'objets évolués
- la modélisation-simulation dans le domaine des problèmes de gestion ?

Peut-on extrapoler cette démarche dans la didactique de l'informatique afin de la mettre au service de la rénovation pédagogique ?

Une démarche d'atelier : montrer et produire

Le département de Gestion et d'Administration des Entreprises de l'I.U.T. de Montpellier tentera de répondre par l'affirmative en présentant **lors d'un atelier**, la démarche pédagogique mise en oeuvre.

- Analyse de dossiers réalisés par des étudiants de 2^{ème} d'I.U.T.
- Construction de savoirs sous Excel 4 ou 5

Deux exemples :

Le dossier de Paie.

L'analyse d'un Seuil de rentabilité dans le contexte du contrôle de gestion.

A - Matériel nécessaire pour le déroulement de l'atelier

- 1 équipement de micro-ordinateurs de type 386 ou 486 avec 4 ou 8 Mo de mémoire dotés du tableur Excel (version 4 et/ou 5)
- 1 poste professeur équipé si possible d'une tablette de rétro-projection
- 1 rétro-projecteur classique
- 1 tableau effaçable à sec et ses feutres.

B - Déroulement de l'atelier

1° Présentation de la méthode pédagogique suivie

2° Analyse critique de dossiers d'étudiants de 2^{ème} année d'IUT de Gestion (1994-1995).

C - Aperçu de la démarche pédagogique

De la finalité pédagogique recherchée par le professeur d'informatique durant l'atelier découlent deux stratégies possibles de travail :

a) On souhaite mettre *l'accent sur tel ou tel concept* d'analyse, d'algorithmique, de programmation, etc. : dans cette hypothèse, le projet proposé sera construit à partir d'un énoncé traditionnellement élaboré voire pré-solutionné...

b) On entend proposer une *véritable démarche de projet* dans laquelle l'étudiant aura à constituer l'ensemble des éléments de sa problématique. Cette hypothèse qui sera la nôtre implique :

- une organisation du travail des étudiants en binôme voire en trinôme.
- la communication d'un thème où n'est guère proposé que le cadre général de l'étude...

Exemple :

Thème: réalisation d'une paie dans une PME du domaine tertiaire comportant 20 salariés cadres et non cadres

Démarche à employer :

1° Description précise de l'environnement - type d'entreprise, caractéristiques, organisation, moyens informatiques disponibles dans l'entreprise, etc ...- (explicitation de "l'énoncé du problème").

2° Etude de faisabilité (analyse fonctionnelle)

3° Etude logique :

- cahier de charges (ergonomie du logiciel à réaliser)
- règles de gestion
- analyse conceptuelle modulaire (diagramme d'analyse descendante)
- algorithmique des modules

4° Programmation des modules

- traduction des algorithmes (boîte à outils et macro-programmation Excel)
- jeux d'essai

5° Notice utilisateur.

Traduction par l'exemple :

On montrera ci-dessous quelques éléments indicatifs d'un dossier d'analyse et de programmation d'une étude de la Paie procédant de la démarche décrite précédemment.

ANALYSE LOGIQUE : ALGORITHMES

On trouvera ci-dessous quelques éléments de l'analyse descendante permettant de comprendre la démarche algorithmique proposée ...

PROGRAMME PRINCIPAL [MENU]

Début

Initialiser *choix* à zéro

Tant que *choix* < 5 (fin de travail)

 écrire (menu)

 lire (*choix*)

 si *choix* =1 alors procédure FICH()

 si *choix* =2 alors procédure BULTYPE()

 si *choix* =3 alors procédure PAIE()

 si *choix* =4 alors procédure EDIT()

ftq

Fin

SOUS-PROGRAMME [FICH]

Début

Ouvrir le fichier base de données "employés"

consulter ou modifier le fichier

enregistrer éventuellement les modifications

fermer le fichier "employés"

Fin

SOUS-PROGRAMME [BULTYPE]

Début

Ouvrir le fichier "bulletin de paie-type"

consulter ou modifier le fichier

enregistrer éventuellement les modifications

fermer le fichier "bulletin de paie-type"

Fin

Etc ...

SOUS-PROGRAMME [PAIE]

Début

Ouvrir un fichier nouveau document "bulletin de travail"
le sauvegarder
ouvrir le fichier "bulletin de paie-type"
sélectionner l'ensemble des informations
recopier la sélection sur le "bulletin de travail"
ouvrir le fichier "employés"
lire (n° employé)
extraire l'enregistrement correspondant :
 (nom et prénom)
 (adresse rue)
 adresse ville)
 (code postal)
 (n° insee)
 (qualification)
 (emploi)
recopier l'enregistrement sur "bulletin de travail"
lire(date fin)
lire(NBH normales)
lire(NBH sup 25%)
lire (NBH sup 50%)
lire (primes)
lire(indemnités)
activer les calculs de paie de "bulletin de travail"
fermer les fichiers ouverts

Fin

SOUS-PROGRAMME [EDIT]

Début

Ouvrir le fichier "bulletin de paie"
lire(imprimer O/N ?)
si choix =O alors procédure (IMP)
si choix = N alors consultation à l'écran

Fin.

Robert WALD

Université Montpellier II, IUT de Montpellier,
 Département G.E.A., 99 avenue d'occitanie 34096
 Montpellier Cédex 5 - France

Les outils de l'informatique: didactique pour l'enseignement secondaire, didactique pour l'enseignement de l'utilisation des logiciels en milieu professionnel, quelles influences?

Sass FERENC

présenté le vendredi 12 avril 1996 à 12h00

Dans le cadre des cours explicitement désignés comme cours d'informatique¹, les finalités d'une formation à une utilisation raisonnée des logiciels dans les établissements de l'enseignement secondaire sont souvent:

- découvrir les bases de l'informatique: matériel, logiciels;
- développer les capacités de résolution de problèmes dans les différents paradigmes engendrés tant par les logiciels que par les langages de programmation, même si ces derniers sont moins fréquemment abordés;
- intégrer les compétences d'utilisation acquises dans les autres disciplines: le traitement de texte pour les notes de cours, le tableur pour un travail en sciences,...

La communauté des professeurs chargés de cette formation réfléchit depuis plusieurs années et tente de définir des pistes méthodologiques pour atteindre ces finalités.

Les finalités d'une formation à l'utilisation des logiciels s'adressant à des personnes dans le milieu professionnel, hors du cadre scolaire², sont avant tout:

- obtenir une efficacité immédiate dans le cadre de son travail.

Deux questions se posent dès lors:

- L'expérience, la didactique, la méthodologie prônées dans l'enseignement secondaire peuvent-elles être transférées dans la formation professionnelle? En d'autres termes, doit-on former les professionnels à une utilisation raisonnée telle que définie pour l'enseignement secondaire, et comment?
- L'expérience de la formation professionnelle peut-elle nourrir les réflexions de l'enseignement secondaire? Avant tout au travers des situations problèmes créées durant l'utilisation quotidienne des logiciels.

L'objectif de l'exposé est d'essayer de montrer quelques éléments de réponses à ces questions, à l'aide de nombreux exemples pratiques.

1-Les cours à orientation professionnelle tels que traitement de texte en bureautique ou d'autres, ne rentrent pas dans cette catégorie.

2-Il s'agit des personnes travaillant dans des bureaux en tant que secrétaires, responsables divers,... Leur âges et formations étant très variables.

DIDACTIQUE POUR L'ENSEIGNEMENT SECONDAIRE DIDACTIQUE POUR L'ENSEIGNEMENT EN MILIEU PROFESSIONNEL QUELLES INFLUENCES?

LES PUBLICS CIBLES

Les cours d'informatique en Belgique, sont organisés dans différentes options à raison de 2 à 4 heures d'informatique par semaine. Les finalités de ses cours ont relativement peu évolué dans le temps. Leurs mises en oeuvre aux travers d'objectifs opérationnels a, par contre, fort changé: il est connu qu'en moins de 10 ans, le cours d'informatique a évolué de l'enseignement de la programmation vers l'utilisation raisonnée des logiciels en passant par le tandem algorithmique-programmation.

De tous les changements intervenus durant cette période durant laquelle j'ai enseigné dans ces options, je ne retiendrai que quelques principes:

- en programmation, il faut éviter les programmes spaghettis, et inciter les élèves à pratiquer une analyse structurée et descendante dans le cadre de résolution de problèmes (sans être dogmatique);
- les élèves ayant pratiqué l'algorithmique et la programmation arrivent généralement plus facilement à utiliser les logiciels de manière raisonnée,
- pour des élèves n'ayant pas de connaissances et compétences en programmation, il faut éviter de les transformer en presse-bouton et clicqueurs de souris. Il faut les aider à développer leurs capacités cognitives en les confrontant à des résolutions de problèmes même avec les logiciels-outils,

Au fur et à mesure que l'ergonomie des logiciels s'améliore, le temps libéré pour l'acquisition des capacités instrumentales devrait être consacré à l'augmentation des capacités cognitives à l'aide de problèmes plus complexes.

Depuis plusieurs années, je m'occupe de la formation des enseignants à l'utilisation des logiciels ainsi que de la formation et de l'accompagnement d'une équipe de secrétaires dans les bureaux de la fédération de l'enseignement secondaire catholique francophone de Belgique.

Les deux publics, élèves et personnel professionnel sont évidemment différents. Le tableau ci-après est une tentative de comparaison entre les deux publics.

Elèves	Personnel professionnel
Le cours d'informatique est optionnel et théoriquement librement choisi.	L'utilisation de l'outil est souvent imposée par la hiérarchie, souvent sans consultation préalable du personnel sur le type d'outils.
L'âge des élèves se situe dans un intervalle réduit (2à3ans).	La pyramide des âges est très large, de la jeune adulte sortant des études à la personne proche de la pension.
Les connaissances « informatiques » sont à des niveaux très variables.	Les connaissances informatiques quasi nulles ou alors acquises au travers de l'utilisation quotidienne des logiciels. Très peu de formations reçues sauf les plus jeunes durant leur études.
L'informatique est perçue comme ludique, facile, moderne, utile ... Le contact avec l'outil dépend de l'équipement personnel de l'élève: 1, 2 heures/semaine sil n'est pas équipé, sinon très variable. Certaines capacités instrumentales sont vite acquises (brancher, éteindre, manipulation souris,...) D'autres ne le seront peut-être jamais (clavier correctement utilisé, ...)	Les logiciels sont des outils de travail. Certain personne travaille 6 à 8 heures d'affilée avec les logiciels. Les capacités instrumentales sont rapidement acquises par l'exercice.
Les travaux sont proposés par le professeur ou sont des projets des élèves. La finalité est la formation des élèves.	Les travaux sont demandés par des supérieurs dont le soucis est l'efficacité dans la gestion quotidienne.
Les élèves sont souvent évalués par le professeur dans le cadre plus large des conseils de classe.	Le travail doit être fait, le mieux et le plus rapidement possible, selon certains chefs;
Les problèmes à gérer le sont rarement dans l'urgence.	Les problèmes rencontrés devraient être résolus dans la minute.
Le niveau de difficulté des problèmes est géré par l'enseignant, en respectant une progression adaptée au élèves.	Les problèmes posés par les utilisateurs sont de niveaux très variables et imprévisibles, nécessitant parfois des compétences supérieures.
La formation suit une progression établie sur 6 années de cours, avec ces finalités, buts, ... définis dans un cadre pédagogique.	Dans beaucoup d'entreprises, la formation est souvent inexistante. Les logiciels sont installés et le personnel doit se débrouiller. A la fédération, la formation initiale est accompagnée de formation de mise à niveau. Un accompagnement quotidien est assuré par le formateur et ses collègues.
La motivation des élèves doit être maintenue, mais ils en ont pour 6 ans maximum.	La motivation du personnel est très variable, et pour certaines personnes, l'utilisation de l'outil peut devenir une contrainte lourde à assumer durant des années de carrière.

LES FORMATIONS

Leur contenu

Le programme du cours d'initiation à l'informatique dans l'enseignement secondaire est relativement bien structuré et il est normal que certains éléments de ce programme soient directement repris dans le cadre de la formation du personnel professionnel. La finalité essentielle est de donner au personnel une image mentale correcte des outils, en vue d'une utilisation la plus adéquate et raisonnée des logiciels:

- La description hardware de l'ordinateur
- Le traitement formel de l'information
- Le système d'exploitation: son rôle, ses fonctionnalisés
- Les fonctionnalités de base des logiciels

Cependant, la nécessité de mettre en oeuvre des fonctionnalités avancées des logiciels apparaît très rapidement. De plus, d'autres exigences et demandes de compétences sont pratiquement imposées: par exemple, la mise en page correcte des textes, l'intégration de dessins, travail sur un réseau, ...

La méthode secondaire appliquée au professionnel

Si le contenu des formations peuvent avoir des points communs, qu'en est-il de la méthodologie?

Pour les deux publics, c'est une méthode s'inspirant de la pédagogie par la résolution des problèmes qui a été adoptée.

L'utilisateur doit percevoir chaque logiciel comme un exécutant formaliste avec lequel, par lequel il doit faire faire une tâche donnée, représentant le problème.

De manière caricaturale, il s'agit de faire passer les formés de l'état de courageux méthodique à l'état de paresseux raisonné³.

Pour un public professionnel, l'adhésion à ce type d'approche plus théorique n'est pas immédiate, car elle ne donne pas l'impression de régler les problèmes immédiats. Elle est acceptée parce qu'elle est suivie d'un accompagnement quotidien. En dehors des moments de formation scolaire, le formé sait qu'il peut compter sur le formateur pour l'aider à résoudre les problèmes urgents. Des formations régulières de mise à niveau sont prévues, d'une part pour partager

³ Le courageux méthodique se reconnaît immédiatement; il est prêt à faire un maximum d'actions à la main: depuis dépalcer le point d'insertion en maintenant le doigt enfoncé sur une touche jusqu'à refaire (à la main) toujours le même mise en page des titres. Le paresseux raisonné est celui qui fait faire un maximum de travail par le logiciel en allant éventuellement jusqu'à la programmation.

l'expérience acquise (tout le monde n'a pas eu à résoudre les mêmes problèmes), d'autre part pour aborder de nouveaux concepts.

La résolution des problèmes urgents est effectuée dans l'esprit de la théorie: le formateur ne donne pas de solution, même s'il en a une, mais aide la personne à construire sa solution en se basant sur les acquis et en montrant la nécessité de compléter ces derniers. On évite ainsi la formation presse-bouton, trucs et ficelles, même si à termes, les manipulations instrumentales sont complètement intégrées grâce à la pratique quotidienne.

Un accompagnement régulier des personnes permet d'analyser les avantages, les inconvénients d'une méthode dont les effets sont immédiatement perceptibles.

Globalement, les personnes formées sont satisfaites de l'approche, car elles comprennent ce qu'elles font et pourquoi elles le font. L'autonomie acquise par rapport au spécialiste apporte une valorisation dans le travail. Le spécialiste intervient quand le formé est bloqué; le but est d'augmenter au maximum le seuil des compétences de chacun. La réflexion la plus fréquente est C'est toi qui nous a formé à devenir des paresseux raisonnés, et à exiger que ce soit le logiciel qui travaille et pas nous. Donc, est-ce qu'il sait faire ça et ça, ...

La méthode ne semble pas permettre de faire franchir **par tous**, des niveaux de difficulté. C'est comme si certaines personnes refusaient ou ne pouvaient pas dépasser un seuil de capacité cognitive.

« Après une formation sur les principes les plus élémentaires concernant les bases de données (mono-fichier), une personne me dit: c'est bien, mais ce n'est pas pour moi, je ne travaillerai jamais avec cet outil »

La conception et l'utilisation des macro-instructions a été abordée durant les formations. Le réflexe de créer une macro-instruction n'est cependant pas ancré.

Les difficultés non liées à l'outil informatique, mais plutôt aux compétences de base en arithmétique, français deviennent des obstacles: calculer avec les pourcentages, différence entre copier, déplacer...

Un des dangers inhérent à l'introduction des outils réside dans l'explosion des tâches demandées: en moins de 15 ans, une secrétaire est passée de la présentation de texte dactylographié au texte mis en page à l'aide de traitements de texte proches de la PAO, intégrant textes, listes, tableaux, graphiques, dessins, Cette intégration nécessite une augmentation importante des connaissances dans des domaines particuliers et des compétences cognitives.

La méthode professionnelle appliquée au secondaire

Si les influences décrites précédemment illustrent essentiellement les influences d'une pratique d'enseignement du secondaire, il est légitime de rechercher les influences d'une pratique professionnelle sur l'enseignement secondaire.

- Un des apports d'une pratique quotidienne dans un milieu professionnel réside dans le volume des situations-problèmes rencontrées. En d'autres termes, il est possible d'élaborer une longue liste d'exercices concrets. Ces exercices serviraient ensuite comme support d'apprentissage. Un début de liste est d'ailleurs présenté dans ce document.

- L'utilisation d'une telle liste serait une voie possible pour développer les compétences des élèves. Une autre voie consisterait à mettre les élèves dans des situations d'urgence telles qu'on les rencontre dans les milieux professionnels; ce au travers de leurs projets personnels suffisamment complexes à leur niveau. Certaines situations-problèmes pourraient d'ailleurs être menées à terme à l'aide de programmes écrits avec le langage des logiciels.

- La relation enseignant-enseigné-matière dans le milieu professionnel est moins pervertie par l'évaluation que dans l'enseignement. Créer une relation d'accompagnement, de collaboration tout en maintenant l'évaluation est manifestement une idée à développer dans l'enseignement secondaire.

- L'utilisation des outils-logiciels actuels implique souvent une connaissance élémentaire de mise en page. Il est souvent demandé aux professionnels de bien présenter les documents imprimés, quelles que soient leurs origines. Il paraît essentiel de sensibiliser les élèves du secondaire à cette problématique, mais il n'est pas du tout certain que cette sensibilisation doit être parmi les finalités d'un cours d'informatique. On ouvre une porte vers la multidisciplinarité, mais c'est un autre débat.

- La gestion du temps a une influence fondamentale sur l'évolution de l'apprentissage. On peut imaginer une classe dans laquelle les élèves utiliseraient intelligemment les logiciels comme outils au service de leur apprentissage durant toute la journée; les professeurs, y compris celui d'informatique seraient à la disposition des élèves pour les faire progresser dans leurs apprentissages et inversement.

CONCLUSIONS

La construction des savoirs au travers de la résolution des problèmes est une des méthodologies adaptées non seulement au milieu scolaire mais aussi au milieu professionnel. Les apprentissages presse-bouton n'ont pas tellement la faveur des professionnels.

Il n'est pas certain que la pratique personnelle des enseignants dans l'utilisation des logiciels soit toujours génératrice de situation-problème complexe.

La gestion des problèmes quotidiens dans le milieu professionnel donnerait naissance à une banque d'exercices qui permettraient d'augmenter le niveau de

difficulté des situations-problèmes rencontrées par les enseignants et élèves, jusqu'à leurs proposer la conception de modules de programmation.

Une utilisation plus intense des outils-logiciels devrait être envisagée dans le cursus des élèves, si possible en collaboration avec tous les enseignants des différentes disciplines. Les professeurs d'informatique y retrouverait probablement le rôle qu'ils pensent parfois avoir perdu.

Sass FERENC

Chargé de mission (informatique et pédagogie) à la F.E.Se.C.

Formateur du personnel du S.E.G.E.C. à l'utilisation des logiciels

BIBLIOGRAPHIE

- Aho, Ullman**, Concepts fondamentaux de l'informatique. Dunod
- Anciaux J.P.**, Les règles d'or de la communication écrite. Les éditions de l'organisation.
- Collins Betty**, Tehnology enriched Schools. STE Publications (Expérience ACOT aux Etats-Unis)
- Date An introduction to DataBase System. Adison Wesley
- Duplan, Jauneau**, Maquette et mise en page. Edition du moniteur
- Fiches pédagogiques du CEFIS. FNDP
- Microsoft Excell 5.0 Manuel de référence
- Microsoft Word 6.0 Manuel de référence
- Microsoft Foxpro 2.6 Manuel de référence
- Novell Wordperfect 6.1 Manuel de référence.
- Papert S.**, l'enfant et la machine à connaître. Dunod
- Parker et Thérien**, Mise en Page. Reynald Goulet
- Sass F. Vandeput E.** Informatique Utile et Raisonnée, tome 1 à 4. Van In.
- Tinsley D., Van Weert J.**, WCCE 95, Liberating the Learner. Chapman & Hall (actes du colloque)
- WordBasic Manuel de référence
- WordPerfect Programming Langage Manuel de référence

ILLUSTRATION DE LA METHODE PAR L'EXEMPLE

Les quelques énoncés repris ci-après sont tous des énoncés de tâches réelles, demandées et effectuées par des personnes ayant suivi les formations. Leur énoncé amélioré peut convenir pour alimenter des banques d'exercices destinées aux élèves de l'enseignement secondaire. L'objectif est avant tout de faire résoudre un problème, la connaissance des logiciels en découlera naturellement.

Les énoncés des tâches s'adressent à des personnes qui ont acquis des compétences avancées dans l'utilisation du traitement de texte et des compétences de base dans l'utilisation des gestionnaires de base de données:

- Gestion du texte: copier, déplacer, effacer, sélectionner une partie du texte, enregistrer, ouvrir, imprimer
- Mise en page: changement de polices de caractères, choix des dimensions et orientations des papiers, gestion des marges, des retraits de paragraphes, mise en colonnes, insertion de dessins et de graphiques, placement de filets, tableaux
- Automatisation du travail: numérotation des pages, pied, tête de page, rechercher, remplacer, publipostage ou mailing, feuilles de styles pour la mise en page
- Notions de tables relationnelles: enregistrements, champs, informations et types, créations et modifications de structures
- Création d'états imprimés: tête et pied de page, regroupement des données, calculs, comptage, ...
- Expression de requêtes: expressions logiques, opérateurs logiques

Les logiciels utilisés sont:

- Traitement de texte: WordPerfect 5.1 DOS et 6.1 Windows.
- Base de données: FoxPro DOS et Windows.
- Tableur: Excell (formation en cours et non encore exploitée)

Le choix des marques des logiciels a peu d'influence sur la méthode. A partir de l'analyse de la tâche demandée, on essaye de déterminer des modèles de solutions qui devraient résoudre le problème et on regarde ensuite si le logiciel permet de mettre en oeuvre une ou plusieurs solutions.

La compétence ultime sera de choisir l'outil qui convienne le mieux pour la tâche demandée.

Chaque exemple est présenté en trois parties:

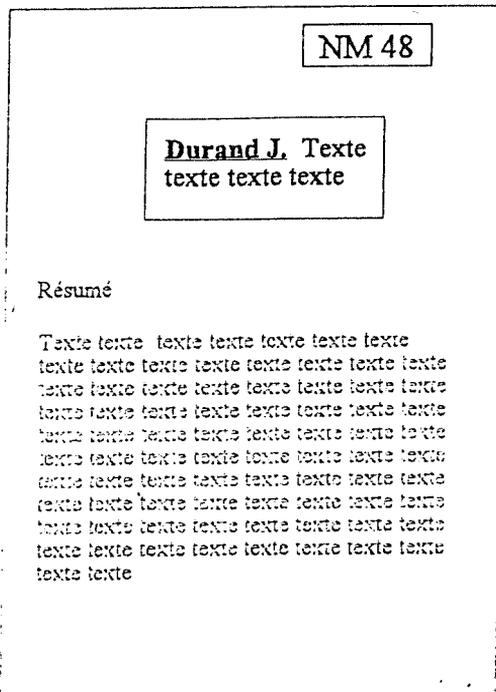
- L'énoncé de la tâche. Cet énoncé n'est jamais formulée par le formateur, mais souvent par une personne extérieur qui s'adresse à la secrétaire ou par la secrétaire en disant: il faudrait faire ça!

- Une partie blanche pour permettre au lecteur qui le désire de réfléchir à sa propre approche du problème et ne pas lui montrer directement ma proposition.
- La proposition que j'ai faite pour répondre à la demande. Ces propositions sont des exemples dont le seul mérite est d'avoir été expérimentés. Ils servent à illustrer une méthode et non pas la solution au problème posé.

L'analyse de chaque énoncé est soumis au principe suivant: éteindre l'ordinateur, ce dernier étant souvent un élément perturbant durant la phase d'analyse..

TRAITEMENT DE TEXTE 1

Enoncé: le travail demandé est imprimer une bibliographie sur des feuilles A5 présentée de la manière suivante:



Le nombre de pages de la bibliographie dépasse 200.

La secrétaire vient me trouver après avoir encodé 5 pages, en disant que c'est un travail de fou, et me demandant si j'avais une solution à son problème: elle ne voulait pas continuer comme cela.

Quelle serait votre proposition?**Ma proposition**

Analyse du travail effectué: encodage des pages une à une, en reprenant par copier/coller sur chacune les encadrés etc.... Le travail est long, fastidieux et décourageant.

La demande est faire plus vite et plus agréablement.

L'analyse du problème

Il est difficile de reprendre en détail les questions et réponses qui ont amené progressivement la secrétaire à trouver une solution qui a été mise en oeuvre. Voici les étapes principales de la démarche:

Observation de la répétition:

toutes les pages sont identiques dans leur présentation. L'idée de créer une page et de la recopier a été envisagée mais abandonnée pour des raisons évidentes de mise en oeuvre pour un document aussi volumineux.

Alternative: voir dans le document demandé un modèle de page dans lequel viennent se placer des informations différentes

Question: est-ce que dans l'univers des traitements de texte, il y a une fonctionnalité qui corresponde à cette alternative ⁴ ?

Réponse: la fonctionnalité la plus proche semble être le mailing qui permet d'effectuer la fusion entre un document type et les champs d'un fichier de données. Comme le mailing était utilisé pour des lettres type avec des noms, adresses, ... il n'était pas évident de l'envisager pour des champs contenant des textes longs avec graphique, tableaux etc. ... Le logiciel de traitement de texte utilisé permet de mettre autant de paragraphes, de tableaux de graphiques que l'on voulait dans les champs des enregistrements, donc la solution pouvait être envisagée.

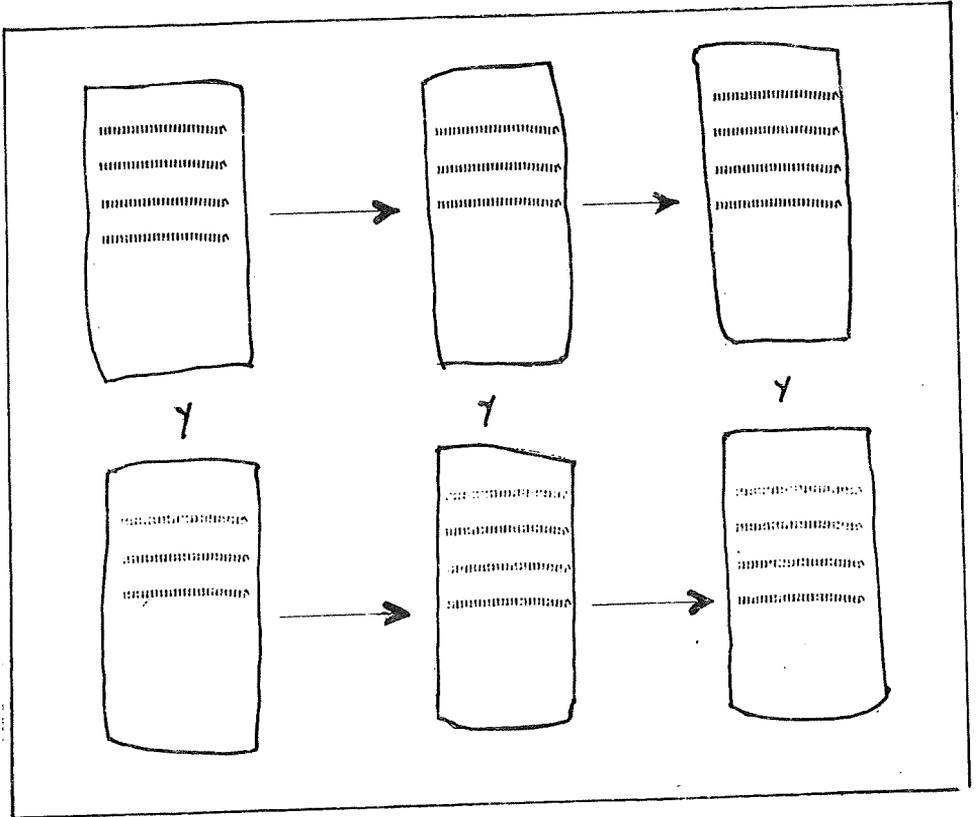
Solution: créer un document type reprenant la mise en page d'une page à l'aide d'une feuille de styles. Créer un fichier de données (200 enregistrements et plus) dans lequel les champs des enregistrements contiendraient le texte, même long et éventuellement mis en page. (Les connaissances acquises à propos des fichiers à joué un rôle non négligeable dans la compréhension) Demander le mailing entre ces deux documents. La mise en oeuvre instrumentale n'a plus posé de problèmes.

Encore plus loin: non prévu dans la demande initiale, l'intérêt de pouvoir sélectionner, trier les informations dans le fichier de données a été très apprécié. Sans parler de la mise à jour des informations. Tout cela uniquement à l'aide d'un traitement de texte.

TRAITEMENT DE TEXTE 2

Enoncé: Lors d'un passage dans un bureau, une secrétaire, que je n'ai pas eu en formation et que je n'avais jamais rencontrée, me montre un document manuscrit qu'elle a reçu des mains de son directeur.

⁴ A cette époque la notion de modèles de document tel qu'envisagée dans les logiciels récents (Word Wordperfect sous Windows) n'existait pas



Regardez, le chef me demande de faire ce document à l'aide du traitement de texte. On m'a dit que vous êtes un spécialiste. Dans quel menu dois-je aller pour le faire?

Votre proposition ?
Ma proposition

Je me retrouvais dans une situation particulière: je ne connaissais ni les compétences de la personne, ni les limites du logiciel (Word 3.0 sur MacIntosh). L'hypothèse de départ était que: seul le traitement de texte pouvait être utilisé.

Analyse:

J'ai demandé à la personne de me dire ce qu'elle voyait dans le document et si possible de différentes manières.

La première perception: je vois 6 cadres dans lesquels il y a du texte et des dessins de flèches.

Les autres perceptions: un tableau, des cadres dans des colonnes, des flèches entre, ...

L'étape suivante a consisté à choisir parmi les perceptions, celle qui allait être mise en oeuvre en fonction de différents critères: facilité de la mise en oeuvre et de la mise à jour éventuelle, possibilités du logiciel, exigence de précision, ...

Le problème des flèches a été réglé en utilisant des caractères typographiques ornementaux. L'utilisation d'un logiciel de dessin a été rejetée par la personne.

La solution retenue par la personne a été celle du tableau. Elle savait comment gérer les tableaux avec son traitement de texte. Durant la mise en oeuvre, suite à toute la préparation, elle a été amenée à découvrir d'autres possibilités et limites du logiciel.

La recherche du menu, de la touche n'a dès lors plus été problématique. Elle a trouvé toute seule.

GESTIONNAIRE DE FICHIERS

Enoncé: une secrétaire a construit une table avec un certain nombre de champs, dont un contient un nombre représentant aussi bien un montant négatif que positif (champ montant) et un autre contenant une information représentant un code postal (champ codepost). Elle a fait trier les enregistrements sur les champs montant et codepost et elle ne obtient pas ce qu'elle veut, c'est-à-dire tous les montants négatifs triés par codes postaux, suivis de tous les montants positifs triés par codes postaux.

Résultats obtenus

Montant	Codepost
-1478	3000
-156	1400
-156	4000
150	6600
268	1742
1000	6500
1457	2500
2457	2350
2536	4780

Résultats souhaités

Montant	Codepost
-156	1400
-1478	3000
-156	4000
268	1742
2457	2350
1457	2500
2536	4780
1000	6500
150	6600

Votre proposition

Ma proposition

La secrétaire avait des compétences suffisantes pour créer la structure des données, trier les données, créer des états imprimés, ... et elle était persuadée que son raisonnement était correct: en faisant trier sur les champs montant et codepost, elle devait obtenir le bon résultat.

La difficulté d'observation et d'analyse a été créée par le contenu de la table elle-même: tous les montants étaient différents et donc le regroupement provoqué par le tri n'était pas perceptible. Il a fallu rappeler cet effet de regroupement, en analysant une table triée sur une expression combinant deux champs (des branches inspectées par des personnes). Une simulation sur d'autres situations a permis d'observer que le logiciel avait bien trié les données correctement.

Sil y avait eu 10 fois le montant 1456, on aurait observé qu'à côté des 1456, les codes postaux étaient bien dans l'ordre croissant.

L'utilisation des capacités du logiciel ne posait aucun problème, mais bien l'analyse de la tâche posée. Le problème restait donc entier.

L'analyse

De questions en réponses, la secrétaire arrive à la conclusion que le montant ne devait pas intervenir dans le tri, mais uniquement son signe et le code postal. Mais le signe était dans le montant. Le logiciel permet-il de ne prendre en charge que le signe. Une recherche dans les possibilités du logiciel donne une réponse affirmative. Le problème était résolu.

Plus loin

Il était intéressant d'aller plus loin dans l'étude de ce problème et de faire apparaître d'autres possibilités. Et si le logiciel n'avait pas permis ce type de manipulation? La réponse ne s'est pas fait attendre: il faudrait une information concernant le signe du montant, donc un nouveau champ dans la structure. L'encodage des valeurs de ce champ pouvait même être automatisé en utilisant les fonctions de remplacement. Pour illustrer la méthode, cette approche a été mise en oeuvre, même si une autre solution existait déjà.

QUELQUES EXERCICES NON-RESOLUS

Les énoncés d'exercices proposés ci-après proviennent tous de problèmes concrets rencontrés dans la pratique quotidienne. Les propositions de solutions, de méthodes (sans utiliser la programmation pure) peuvent être envoyées à Sass F., Mande Saint-Etienne 923, 6688 Longchamps.

Traitement de texte

Les énoncés sont proposés en tant que documents à réaliser dans les pages suivantes. Ils ne peuvent être réalisés qu'à l'aide d'un traitement de texte.

Document 1

Obtenir le document suivant sur une page A4, uniquement avec traitement de texte.

François Sass
Notaire
Rue des Actes, 18
6688 Payons

Monsieur le client
adresse
code postal localité

COMPTE ACQUEREUR

Acte fixé le

Débit	
Prix d'achat ou solde	
Frais d'acte d'achat	
Frais d'acte de prêt	
Indemnité d'occupation	
Quote-part foncier 19	
Frais de plan	
Quote-part acte de base	
Divers	
	Total
Crédit	
Acompte versé	
Dépôt en l'étude	
Montant net du prêt	
Prorata de loyer	
Indemnité d'occupation	
Divers	
	Total
Différence à combler	

Somme dont vous voudrez bien vous munir, en espèces, le jour de la passation de l'acte.

Document 2

Ce document est un cas particulier: il doit être obtenu par programmation, en utilisant le langage intégré au traitement de texte. Il représente un agenda annuel dont les feuilles ont les dimensions 9,9 cm. x 16 cm.

Novembre 1995		
Me 1		
J 2		
V 3		
S 4		
D 5		
L 6		
M 7		
Me 8		
J 9		
V 10		
S 11		
D 12		
L 13		
M 14		
Me 15		

J 16		
V 17		
S 18		
D 19		
L 20		
M 21		
Me 22		
J 23		
V 24		
S 25		
D 26		
L 27		
M 28		
Me 29		
J 30		

Votre Proposition

GESTIONNAIRE DE BASE DE DONNEES

Le niveau de difficulté des problèmes rencontrés est très variable. L'ordre proposé ci-après ne contient aucune progression dans la complexité. Le vocabulaire utilisé est celui du modèle relationnel. Le but de tous les exercices est déviter le réencodage des informations (pratique des courageux méthodiques), en utilisant toutes les fonctionnalités des logiciels, sauf la programmation.

1. On dispose de deux tables contenant un code et un libellé. Les informations contenues dans ces tables concernent le même objet, mais ont été encodées à des lieux différents. Il est demandé de comparer le contenu des deux tables pour faire apparaître les différences.
2. On dispose d'un table contenant un champ dénommé TEL représentant un numéro de téléphone. Les numéros de téléphone ont été encodés sous deux formats

différents: 99/9999999 ou 999/999999 (plus les erreurs d'encodage). Pour des raisons d'harmonisation, il est demandé d'obtenir les numéros de téléphones sous la forme: (99) 999 99 99 ou (999) 99 99 99.

3. On dispose d'une table contenant un champ dénommé NOM représentant un nom de famille suivi de l'initiale du prénom suivi d'un point (Dupont J., Durant J.M., ...). On demande de séparer ces deux informations dans deux champs différents: le champ NOM contenant uniquement le nom de famille, le champ PRENOM l'initiale du prénom.

4. On dispose d'une table contenant les informations suivantes à propos des options organisées dans les écoles: province, école, option, nombre de filles, nombre de garçons. Un enregistrement représente la relation: une option est organisée dans l'école de la province et est suivie par autant de garçons et autant de filles. (Si l'école organise 15 options, il y aura 15 enregistrements pour cette école.).

- Faire imprimer la population totale par province, par école et pour tout le pays.
- Faire imprimer la population totale par option.
- Faire imprimer les écoles organisant plus de 5 options avec moins de 100 élèves.
- Faire imprimer pour chaque option, la liste des écoles l'organisant.

5. On dispose d'une table contenant des informations représentant des personnes ayant des statuts divers pour différentes commissions: directeurs, présidents, membres, secrétaires, ... Il s'agit de faire imprimer, pour chaque commission, les noms, prénoms des personnes mais en respectant l'ordre hiérarchique: président, directeur, secrétaire, les membres.

Sass FERENC

Chargé de mission (informatique et pédagogie) à la F.E.Se.C.

Formateur du personnel du S.E.G.E.C. à l'utilisation des logiciels

Les Rencontres à thème: **Les Recommandations Finales du Colloque**

Le programme du colloque a fait place à deux périodes d'une heure où les participants ont pu échanger en groupes restreints sur les problèmes liés à la place de l'informatique dans l'éducation, à son enseignement et aux diverses pistes suivies dans les pays francophones à ce propos.

Le but des rencontres à thème et leur organisation

L'objectif, outre celui de partager expériences et réflexions, était de produire un poster qui représente une recommandation (ou quelques-unes) qui semblait essentielle à chacun des groupes. Cette affiche devait énoncer une proposition réaliste et préciser pourquoi cette proposition paraît importante et motivée. Les différentes recommandations devaient dès lors prendre une forme commune:

Etant donné que, Nous proposons que.....;
--

A l'issue de la seconde rencontre, les diverses propositions ont été affichées et chaque groupe a répondu aux questions posées par les membres des autres groupes à propos des recommandations énoncées.

Enfin, ces propositions ont également servi à alimenter les débats de la table ronde finale. Six groupes ont été constitués. Ils étaient animés par Mesdames Catherine Couderc, Marie-Thérèse Rey et Chantal Richard et par Messieurs: Abdelhafidh Abidi, Ridha Haj Ali, Charles Duchâteau, Jean-Pierre Peyrin.

Quelques thèmes proposés aux réflexions

Tout en laissant à chacun Des groupes pleine liberté d'aborder les thèmes souhaités (pour autant qu'ils s'inscrivent dans la problématique générale de la didactique de l'informatique), un certain nombre de questions avaient été proposées à la réflexion. Elles sont le fruit du travail de Alain BRON, Charles DUCHATEAU, Raymond MOREL et Jean-Pierre PEYRIN.

Autour de l'évolution des curriculums

- Faut-il encore un enseignement de l'informatique au secondaire? Pourquoi?
- Où en est la révision des curriculums pour les élèves? Et pour les enseignants en formation initiale? Et en formation continue?
- Que penser du statut optionnel d'un cours d'informatique?
- Y a-t-il des éléments neufs en ce qui concerne les rapports entre l'informatique discipline et l'informatique outil pour enseigner et apprendre?
- Quel curriculum pour les premiers cycles universitaires en ce qui concerne l'enseignement de l'informatique à des non futurs professionnels?

Autour de la formation des enseignants

- Quelle formation pour les enseignants utilisateurs "naïfs". Et pour les enseignants chargés du cours d'informatique? Et pour les enseignants personnes ressources au sein de l'école?
- Quel statut pour les enseignants chargés de la logistique en ce qui concerne l'utilisation des NTIC au sein de l'école?

Autour du contenu d'un cours d'informatique

- Quels sont les concepts essentiels de "l'informatique des utilisateurs"? Que nous apprennent à ce propos les dernières versions des programmes d'enseignement des pays francophones?
- Peut-on éviter la dérive qui fait du cours d'informatique un ensemble de modes d'emploi d'instruments logiciels? Comment faire percevoir les notions fondamentales et former des élèves adaptables et créatifs?
- Comment prendre en compte les développements récents : multimédia, Internet...?
- Et la programmation? Que penser de voies nouvelles comme la programmation au sein des logiciels, la robotique, la programmation orientée objet? Quelle attitude en ce qui concerne les cours optionnels où tant les professeurs que les élèves souhaitent continuer à faire de la programmation "à la Pascal"?

Autour des méthodes d'enseignement utilisées dans le cours d'informatique

- Y a-t-il des démarches particulières mises en avant au sein d'un enseignement de l'informatique et qui recèlent un fort potentiel pédagogique, susceptible d'être transposées à d'autres cours ou d'autres situations?
- Un cours d'informatique doit développer chez les élèves à la fois des savoirs et des savoir faire. Quel sont les problèmes spécifiques liés à un enseignement efficace et correct des savoir faire? Qu'évoquent pour vous dans ce contexte les termes "formation balisée et assistée"?
- La recherche et le traitement d'informations sont au coeur de l'informatique des utilisateurs. Quelles stratégies promouvoir pour développer ces aptitudes et ces attitudes dans le cadre d'un cours d'informatique?

La collaboration dans le monde francophone.

- Quelles seraient actuellement les collaborations à promouvoir en ce qui concerne l'enseignement de l'informatique? Les nouveaux moyens de communication (réseau...) sont-ils une opportunité de développer nos échanges? Sur quels thèmes?

Les recommandations des groupes

Voici la transcription sans modification des posters proposés par les divers groupes d'échange:

1. Etant donné

- la place que prennent l'informatique et l'utilisation de l'ordinateur dans les évolutions économiques, sociales et culturelle de la société,
 - les risques que présenterait un accès inégal à la connaissance et à la culture
- nous proposons que**

• *Un savoir minimal relatif d'une part au traitement de l'information et d'autre part à l'éthique de l'informatique doit être dispensé à tous les élèves dans le cadre de la scolarité obligatoire.*

2. Etant donné que

• la notion de culture générale en informatique doit être précisée en vue de son introduction dans l'enseignement et la formation

nous proposons que

• *les aspects suivants soient pris en compte dans la scolarité obligatoire Des élèves:*

- *éthique et enjeux sociaux de l'informatique*

- *ensemble minimal ,des concepts, des connaissances et des savoir-faire en informatique nécessaire à une utilisation consciente et raisonnée.*

3. Etant donné que

• l'algorithmique relève de la culture générale,

• une connaissance même élémentaire de l'algorithmique permet de programmer des applications performantes,

• l'informatique est une filière professionnelle importante,

nous recommandons

- *Le maintien d'une formation à l'algorithmique de base à travers des environnements adaptés (macro-commande, robotique...).*

4. Etant donné que

• le flou dans lequel nous naviguons depuis plusieurs colloques AFDI à propos du terme « informatique des utilisateurs »

nous recommandons

- *une réflexion et d'une concertation au sein de la francophonie en vue de préciser ce que recouvre ce terme et quelles sont les incidences pédagogiques liées à cette définition.*

5. Etant donné que

• l'utilisation générale d'une série de logiciels horizontaux (traitement de texte, gestionnaire de communication...) et leur intérêt pédagogique,

nous recommandons

• *concertations et échanges sur les modalités de leur apprentissage (créatif et raisonné) et leur enseignement.*

6. Etant donné que

• l'enfant d'aujourd'hui est confronté à l'informatique et que son utilisation sans formation est une idée fausse,

nous proposons que

- *L'école assure une "alphabétisation à l'informatique le plus tôt possible qui permette, entre autres, l'usage raisonné des moyens informatiques.*

7. Etant donné que

• l'informatique est mentalement formatrice,

nous proposons que

• *au même titre que le "lire", "écrire", "compter", l'école est amenée à assurer une formation de base de l'informatique.*

8. Etant donné que

- les expériences des différents pays sont multiples,
- nous proposons**

- pour le prochain colloque de distribuer un document présentant le programme informatique de chaque pays.

9. Etant donné que

- l'informatique touche tous les secteurs du savoir,

nous proposons

- La formation de tous, centrée sur le projet de l'élève sans l'encapsulation dans un cours, mais dans le cadre d'une animation sur toutes les disciplines y compris l'informatique.
- La formation de tous les futurs enseignants quelle que soit leur disciplines.

10. Etant donné que

- le silence de l'AFDI entre deux colloques et l'augmentation de l'effectif Des pédagogues soucieux de l'informatique,

nous proposons

- une rencontre annuelle,
- un groupe AFDI par pays pour poursuivre le travail entre deux colloques,
- de faire participer davantage les pédagogues concernés,
- un serveur WEB,
- une veille pédagogique.

11. Etant donné que

- les considérations philosophiques ne font guère avancer la recherche en didactique,
- le traitement et la recherche d'information sont au coeur de l'informatique des utilisateurs,
- tous les acteurs impliqués dans la formation doivent être interpellés pour assurer une cohérence,

nous proposons que

- le prochain colloque soit consacré à un thème unique et transversal: « LES SYSTEMES D'INFORMATION ».

Le bilan

Au delà des sensibilités diverses et des formulations variées, deux préoccupations essentielles se font jour, semble-t-il, au sein de la communauté des enseignants et chercheurs préoccupés de l'enseignement de l'informatique:

La nécessité d'une alphabétisation à l'informatique et à un usage raisonné des qu'elle a secrétés, destinés à tous les élèves.

Le besoin d'échanger réflexions et expériences sur les constituants de « l'inform des utilisateurs » et les modalités de son appropriation par tous les élèves.

Enfin, et ceci ne peut transparaître dans la formulation des recommandations des divers groupes, la majorité des participants s'est plu à souligner l'intérêt de laisser la place, au sein d'une rencontre comme celle de l'AFDI, à de tels moments d'échange. Ils permettent de mesurer, au delà des différences et des expériences diverses, ce qui relie la communauté des chercheurs et enseignants autour du thème de la didactique de l'informatique. (rapport proposé par **Charles DUCHATEAU**)

La table ronde de clôture

A l'issue de la rencontre, une table ronde a été organisée. Présidée par Tahar HAFIAED, elle réunissait Charles DUCHATEAU, Raymond MOREL, Claude PAIR, Jean-Pierre PEYRIN et Jean-Marie ROULLER.

Après une présentation par Tahar HAFIAED des objectifs de cette table-ronde, l'introduction en est faite par Jean-Marie ROULLER qui fait entendre l'opinion d'un chef d'entreprise en ce qui concerne les missions de l'éducation en général et de la formation aux technologies de l'information en particulier. (voici, par ailleurs le texte intégral de l'intervention de M.ROULLER)

L'avis d'un chef d'entreprise

Jean-Marie ROULLER

Les questions habituelles de l'adéquation entre la formation scolaire, à tous les niveaux, et les besoins de l'économie se posent également pour l'enseignement de l'informatique.

Les différents documents "suisses" traitent du développement de l'enseignement de cette branche et des réflexions à la base des nouveaux concepts. Peut-être ces derniers sont-ils le fruit d'une collaboration intense entre les pédagogues, promoteurs, et le monde économique, utilisateur ?

Pour les entreprises, l'informatique est un outil indispensable à son fonctionnement, à son contrôle et à son développement. Et comme tout outil, il est l'instrument du meilleur comme du pire, selon les compétences et les intentions de l'opérateur informatique, cet artisan des temps modernes.

Que prévoit la formation officielle de base ?

L'Ordonnance sur la reconnaissance des certificats de maturité (Document 1), précise à l'art. 7:

Le but de écoles préparant à la maturité ...

... donner de solides connaissances fondamentales

... s'exprimer avec justesse et précision

... maîtriser les méthodes du travail scientifique et celles du traitement de l'information

...

Si ces conditions de formation sont remplies, si les connaissances correspondantes sont acquises, les tâches courantes de l'entreprise, c'est-à-dire fondamentales pour son bon fonctionnement, pourront être réalisées correctement.

Or qu'en est-il de la réalité quotidienne ?

L'aspect manipulation des PC, - souvent ludique -, et les connaissances de base des principaux programmes de la catégorie "Office" sont connus. Ce qui

manque, ce sont les *solides connaissances fondamentales* pour les appliquer correctement.

Le traitement de textes

* Tout d'abord l'orthographe ...

La connaissance, la sensibilité et le goût pour cet aspect de la langue écrite se réduisent sensiblement d'un lustre à l'autre. Seule "la vieille génération" des ...généralistes réagit aux horreurs lues, au grand étonnement de leurs auteurs !

Les correcteurs orthographiques ne traitant pas tous les cas, et la reconnaissance vocale fiable ne faisant pas encore partie du matériel standard de bureau, cette méconnaissance génère un manque de productivité dans les entreprises, et souvent pénalise leur image de marque.

* ... Ensuite la présentation

La rigueur, l'esthétique et la cohérence font partie intégrante de la présentation d'un document; son aspect visuel est également important pour l'image de marque de l'entreprise.

La dactylographie classique doit être maîtrisée dès le secondaire I. C'est une connaissance fondamentale.

Dès le secondaire II, l'aspect "document" doit être développé, conformément aux règles d'édition. Le recours aux professionnels de l'imprimerie pour la formation des enseignants est une ressource à développer.

Les tableurs

Ces "machines à produire des tableaux" sont à la fois merveilleuses et dangereuses. Combien de statistiques encore plus fausses sont diffusées parce que des erreurs pas assez flagrantes d'algorithmes non contrôlés ont la vie longue.

Les connaissances en calcul oral et l'appréciation de l'ordre de grandeur des opérations sont faibles chez les jeunes sortis des écoles de commerce et de la maturité, à l'exception des scientifiques peut-être. De ce fait les erreurs d'algorithmes sont plus difficiles à détecter par leurs auteurs.

Au niveau secondaire II, il est donc nécessaire de développer, parallèlement à l'acquisition des connaissances, des techniques de contrôle de plausibilité.

Les bases de données

A la fin de leurs études secondaires, les étudiants doivent être capables de réaliser des bases de données simples. Les mécanismes relationnels seront connus, de même que les concepts d'intégration dans les applications bureautiques.

Aspects qualité

Une approche des processus d'assurance qualité -ISO 9000- tels qu'appliqués dans tous les secteurs de l'économie doivent également être pris en compte dans les plans de formation.

L'enseignement de l'informatique ne pourrait-il pas faire oeuvre de pionnier en introduisant ces concepts à la fois indispensables pour assurer la survie et le développement de l'économie, et prometteurs d'enrichissements personnels par une reconnaissance de la qualité et une formalisation des exigences.

Aspects généraux

Ne pouvant être exhaustif, nous ne retiendrons que deux aspects:

- * la *"justesse et précision"* de l'expression écrite doit être considérée dans l'enseignement de toutes les branches, avec ténacité et sans découragement de la part des enseignants !
- * la *"maîtrise des méthodes du travail scientifique et celles du traitement de l'information"*.

Un effort considérable doit être entrepris pour apprendre aux élèves à aller chercher les informations dans les modes d'emploi et la documentation des producteurs de logiciels et d'applications, quels que soient leurs supports.

Conclusion

Pour l'entreprise, les objectifs de la formation informatique des étudiants doit être conforme aux buts généraux fixés par l'Ordonnance ... maturité, en privilégiant une démarche QUALITE, - solides connaissances de base -, plutôt que le survol d'un monde infini.

L'INFORMATIQUE N'EST PAS UNE PISTE DE
SURF, MAIS UNE MINE DE FOND.

Jean-Marie ROUILLER
ELECTRICITE D'EMOSSON SA
CH-1920 MARTIGNY Monastir, avril 1996

Du fort large débat qui s'est ensuivi, il est impossible de donner un compte-rendu exhaustif. En toute subjectivité, je voudrais souligner les éléments suivants :

- Les recommandations des divers groupes à thème, si elles sont essentielles pour la suite de notre travail de chercheur et d'enseignant, s'adressent pour certaines à l'AFDI, pour d'autres au monde politique. Ces dernières doivent en

tout cas être relayées en direction des décideurs et des responsables du monde de l'éducation de nos divers pays. Il s'agit surtout des propositions suivantes¹ :

-Dans un monde où les technologies de l'information et de la communication sont omniprésentes et envahissent à la fois la sphère professionnelle et la sphère privée, il est impératif que TOUS les élèves aient bénéficié d'une alphabétisation à l'informatique, pendant la période de l'enseignement obligatoire.^{2 3}

-L'idée qu'une utilisation créative et raisonnée des instruments informatiques ne nécessite pas de formation spécifique est fausse.

• En ce qui concerne les propositions qui doivent nourrir les réflexions et recherches à venir des enseignants et chercheurs rassemblés au sein de l'AFDI, on peut mettre en évidence les recommandations suivantes :

-Il est impératif de poursuivre l'analyse et la définition des éléments constitutifs de l'**informatique des utilisateurs** : quels sont les concepts (invariants) que l'enseignement doit aborder ? Quels sont les savoirs et savoir-faire minimaux requis pour une utilisation raisonnée et créative des environnements informatisés ? Il est important de répondre aux questions "Pourquoi ?" (cela rejoint la nécessité des arguments à présenter aux décideurs en matière d'éducation), "Quoi ?" (c'est là tout le problème -récent- de l'exigence d'affiner les concepts à faire maîtriser) et "Comment ?" (où l'on retrouve le besoin de recherches et d'expériences didactiques dans l'appropriation par les élèves de l'informatique des utilisateurs).

-Les échanges et la collaboration entre chercheurs et enseignants de nos divers pays (qui est l'un des rôles essentiels de l'AFDI) doit se poursuivre et s'amplifier.

D'autres aspects ont bien entendu été abordé lors de cette table ronde :

¹Ces propositions sont bien reflétées, par exemple dans les deux recommandations suivantes émanant des groupes de travail :

• Etant donné que l'enfant d'aujourd'hui est confronté à l'informatique et que son utilisation sans formation est une idée fautive, nous proposons que *l'école assure une "alphabétisation" à l'informatique le plus tôt possible qui permette, entre autres, l'usage raisonné des moyens informatiques.*

• Etant donné que l'informatique est mentalement formatrice, nous proposons que *au même titre que le "lire", "écrire", "compter", l'école assure une formation aux notions de base de l'informatique.*

Il est utile dans le dialogue avec les décideurs et au moment de présenter des arguments de garder à l'esprit les divers modèles de l'école, tels que présentés dans la conférence inaugurale de Claude Pair : "L'informatique, science humaine ou inhumaine". Les raisons à invoquer pour une alphabétisation à l'informatique ne sont pas identiques face aux tenants de l'école "républicaine", "domestique" ou "industrielle".

³L'idée n'est pas neuve : "La généralisation dans nos sociétés modernes tant au niveau professionnel qu'au foyer des nouvelles capacités de communication rendues possibles par la micro-électronique conduit actuellement certains éducateurs et certains responsables du développement à se demander si le problème [de l'alphabétisation] ne va pas se poser bientôt dans les mêmes termes pour l'informatique." (DIEUZEDE, H. : Alphabétisation informatique", Actes des Deuxièmes Journées de Réflexion sur l'Informatique, Presses Universitaires de Namur, 1984, pp. 125-128.

- On a souligné la crise de l'école et du système éducatif dans la plupart de nos pays occidentaux : les contributions de Claude PAIR et de Marie-Thérèse REY au cours de ce colloque l'ont bien illustrée.
- L'utilisation des instruments logiciels constituent de merveilleuses occasions d'apprentissage et de résolutions de "vrais" problèmes, significatifs pour les élèves. Ils sont aussi de puissants moteurs de changement du rôle des enseignants et de rénovation pédagogique et cela quelle que soit la discipline enseignée.

La table ronde a été clôturée par une allocution de Monsieur Mohamed Ben AHMED, Secrétaire d'Etat auprès du Premier Ministre Chargé de l'Informatique, qui a donné toute la mesure de l'engagement de la Tunisie dans ce processus de diffusion de la culture informatique au sein du système éducatif.

Rapport proposé par **Charles DUCHATEAU**

EPILOGUE

De l'ordinateur à l'informatique

Conférence préparée par

Jacques ARSAC

pour les 50 ans d'informatique à l'Université de Fribourg
-septembre 1995-

1. La question posée.

"De l'ordinateur à l'informatique" : ce titre, qui m'a été proposé par Alain Bron, pose d'entrée de jeu la question centrale : qu'est-ce que l'informatique ? Si elle est l'ensemble des activités qui mettent en jeu des ordinateurs, il faut montrer quand les conséquences économiques, sociales, voire culturelles des ordinateurs sont devenues telles qu'il a paru nécessaire de tout regrouper sous ce vocable commun.

Si au contraire, comme je l'affirme, l'informatique est une science nouvelle, alors il faut savoir quand et comment on est passé des ordinateurs et de leurs utilisations à la science qui les sous-tend. Mais cette question se double d'une autre, plus actuelle : comment faire passer les élèves des ordinateurs, dont ils ont une perception immédiate favorisée par leur aspect ludique, à la science informatique dont ils sont les instruments privilégiés ?

Dans le cadre d'un cinquantenaire, il me semble que c'est la perspective historique qu'il convient d'aborder. Au demeurant, elle n'est pas étrangère à la perspective pédagogique : elle fournit un chemin effectif des ordinateurs à l'informatique, dont l'enseignant pourra s'inspirer. Les difficultés n'en sont pas pour autant résolues : peut-on faire l'histoire d'un événement dans lequel nous sommes à ce point immergés que l'unanimité n'est pas du tout faite, même dans la communauté scientifique, sur l'existence de cette nouvelle science ? Il y a deux ou trois ans, "L'Institut d'histoire du temps présent" (créé en France par le CNRS) fêtait un anniversaire par un petit colloque. Les médiévistes demandèrent : "comment pouvez-vous faire l'histoire d'événements dont vous ne savez pas s'ils ont une signification historique, faute d'un recul suffisant ?" A quoi les historiens du temps présent répondirent aux médiévistes par une autre question : "Comment pouvez-vous faire une histoire du Moyen Age ? Ce que vous en savez vous est donné par des documents écrits dans une langue qui n'a plus cours. Comment pouvez-vous garantir la signification que vous attachez à ces documents qui ne sont finalement que des suites de lettres ?"

Comme si cela ne suffisait pas, je n'ai pas la compétence voulue pour écrire l'histoire des idées qui amena des ordinateurs à l'informatique. Je vais donc esquiver toutes ces difficultés par un expédient. Je présenterai seulement la façon dont j'en suis venu à l'idée d'une science informatique, à partir de ma pratique des ordinateurs, et des expériences d'enseignement que j'ai faites à partir de 1957. Je serai ainsi amené à traiter ce sujet à la première personne, non parce que je me considère comme important, mais tout simplement parce que je ne peux pas parler au nom d'un groupe de personnes. Il se trouve pourtant que ces quelques pages ne

sont pas les réflexions d'un promeneur solitaire, et que d'autres, notamment des enseignants, se reconnaîtront, je l'espère dans le parcours que je vais évoquer. Peut-être ce témoignage pourra-t-il servir à des historiens du temps présent. Je l'ai déjà esquissé en d'autres circonstances. Mais la persistance du refus de la science informatique par beaucoup oblige à rechercher tout ce qui peut être trop vite dit ou insuffisamment élaboré. Cette version 1995 ne sera pas identique aux précédentes. A l'historien de s'y retrouver dans ces formes successives !

2. Premiers contacts avec l'ordinateur.

Dans les années cinquante, j'ai fait mes débuts de chercheur physicien dans le laboratoire de Radioastronomie de Jean-François Denisse, à l'Observatoire de Meudon. Peu attiré par une théorie trop formelle, pas assez minutieux pour être un bon observateur, je me suis consacré aux problèmes calculateurs de la radioastronomie. Poussé par l'astronome Jean Claude Pecker, je suis allé suivre un cours de programmation pour l'ordinateur IBM 650 en 1956. De cela je suis sûr, parce que j'ai conservé le diplôme de programmeur qui m'a été remis à la fin du stage. J'ai été immédiatement convaincu des services qu'un ordinateur pourrait rendre à l'Observatoire, lieu privilégié des "calculs astronomiques"! J'ai donc entrepris de convaincre la direction de l'Observatoire qu'un ordinateur rendrait par mon collègue Bernard Vauquois, lui aussi chercheur à Meudon. Nous avons donc fait des séminaires pour présenter l'ordinateur aux chercheurs de l'observatoire. Nous leur expliquions que ces machines étaient comparables aux calculatrices électromécaniques de bureau qu'on utilisait à l'époque : accumulateur, quatre opérations, décalages..., avec une réserve de mémoire pour conserver les résultats intermédiaires. Quant aux programmes, ils ressemblaient aux feuilles de consignes que l'on rédigeait alors pour faire exécuter des calculs par les employés d'un service de l'Institut d'Astrophysique à Paris.

Etait-ce par calcul pédagogique, pour prendre appui sur une expérience concrète des chercheurs ? Etait-ce parce que c'était ainsi que Vauquois et moi percevions les ordinateurs à l'époque ? Je ne peux répondre à cette question, et Bernard Vauquois n'est malheureusement plus là pour répondre. Je peux toutefois évoquer un fait que j'ai gardé très présent dans ma mémoire. J'ai suivi peu de temps après le stage de programmation IBM un stage de la compagnie des machines Bull pour la programmation du "Gamma AET". Le nom déjà, à lui seul, est révélateur. Le "Gamma" était un calculateur électronique ayant 64 mots de mémoire. On lui avait adjoint une "armoire extension tambour" (A E T), un tambour de 8000 mots qui servait de périphérique, avec, il est vrai, un système très en avance pour l'époque : pendant qu'on travaillait avec une moitié de la mémoire, on chargeait l'autre depuis le tambour. Un système d'adressage permettant de commuter les deux moitiés. Ce n'est pas cela qui faisait l'essentiel du cour, mais bien plutôt la description du calculateur "Gamma". Il ne s'agissait pas d'un ordinateur au sens moderne du terme, mais d'un calculateur électronique doté d'une extension de mémoire.

3. De nouveaux concepts.

André Danjon, alors directeur de l'Observatoire de Paris, ne fut pas trop difficile à convaincre de faire acquérir par l'Observatoire un ordinateur. Il fallait en faciliter l'accès aux chercheurs. Or c'étaient de petites machines : bien que le 650 IBM qui fut installé à Meudon pesât plusieurs tonnes, il ne possédait que deux mille mots de mémoire, et ne devait pas dépasser au mieux 2000 opérations à la seconde. Le constructeur ne fournissait qu'un assembleur. Bernard Vauquois ayant été nommé professeur à Grenoble, je me trouvais seul pour concevoir un langage adapté à nos besoins et en réaliser le compilateur. Est-ce ce contact avec autre chose que du calcul numérique qui a modifié ma vue des ordinateurs ? Je ne sais...

J'ai été appelé par René De Possel à l'Institut de Programmation qu'il venait de créer à la Faculté des sciences de Paris, en 1964. Notons au passage qu'il ne s'est pas appelé "institut d'informatique". Pourtant le mot existait déjà : il avait été proposé en 1962 par Philippe Dreyfus, dans un séminaire de l'AFCALTI (association française pour le calcul et le traitement de l'information). Mais il ne correspondait à rien, hormis la commodité pour laquelle il avait été forgé : "Quand on me demande ce que je fais", avait expliqué Dreyfus, "je ne sais que récréation de cet institut correspondait à un besoin : il fallait former des programmeurs pour les ordinateurs qui étaient en train de se répandre, dans les laboratoires de recherche comme dans les grandes entreprises. Ces programmeurs auraient à travailler en gestion aussi bien que pour du calcul scientifique.

L'image de l'ordinateur changea. Mon collègue Jean Suchard insistait sur le fait qu'il est d'abord caractérisé par une mémoire, formée de cases dans lesquelles on peut ranger aussi bien des éléments du calcul que des instructions pour la machine. Il se trouve qu'on a appelé "information" ce qu'on met dans une case. J'ai très présent à l'esprit les difficultés qui en résultaient pour moi, et que je m'efforçais de ne pas manifester pour n'avoir pas à en traiter. Dans une case de mémoire, on peut ranger un nombre entier, "réel" (par abus de langage, en fait un nombre en "virgule flottante"), ou une instruction de la machine.

Si je me trompe dans l'écriture du programme, l'ordinateur ira chercher une instruction dans une case où j'ai rangé un entier. Il ne s'en apercevra pas, interprétera cet entier comme une instruction, et le résultat en sera une erreur aux effets imprévisibles. J'avais mis un entier, la machine a lu une instruction. Or on ne peut prendre le sinus que d'un nombre. Qu'y avait-il dans la case, un nombre ou une instruction ?

J'insistais auprès des étudiants sur le fait qu'une case de mémoire est repérée par un numéro, son "adresse", et possède un contenu : ce que l'on met dedans, et qu'on peut consulter ultérieurement. J'utilisais l'image du magasin de chaussures, avec des boîtes sur des rayonnages.

Chaque boîte est reconnaissable par le numéro écrit dessus ; je peux y mettre aussi bien des pantoufles d'homme que des mocassins de femme...

Il ne fallait pas confondre l'adresse d'une case et son contenu. Pas difficile, tant qu'on ne fait pas intervenir l'adressage indirect : on met l'adresse d'un élément du calcul dans une case. Le contenu d'une case est alors l'adresse d'une autre. Il y avait, dans chaque promotion, des étudiants qui avaient du mal à s'y retrouver.

Les premiers grands langages "évolués" étaient disponibles sur les ordinateurs, alors assez puissants pour les supporter. FORTRAN avait été le premier, mais ALGOL 60, mieux construit et défini par une grammaire plus précise, avait la faveur des enseignants. Au niveau de ces langages, il n'était plus question d'adresse et de contenu. Mais les éléments du calcul étaient désignés par des noms (les "identificateurs") et possédaient une valeur. Quand on écrit I ou ALPHA, que met-on en jeu ? le nom, ou la valeur ? FORTRAN avait introduit l'instruction diabolique :

$$I = I + 1$$

Mon collègue Jean Vignes raconte qu'il fit un exposé à des chercheurs de l'Institut Français des Pétroles pour les initier à FORTRAN. L'un d'eux, qui avait été bon étudiant en mathématiques, lui dit à la fin : "Je vous ai bien suivi jusqu'à $I = I + 1$. Mais je n'ai pu aller plus loin" Il fallait donc expliquer aux étudiants qu'un identificateur en partie droite d'une instruction désigne la valeur de l'objet, tandis qu'en partie gauche il désigne le nom. $I = I + 1$ doit être lu comme "à l'objet de nom I donner pour valeur la valeur qu'à l'objet de nom I, augmentée de 1". On arrivait à peu près à faire passer cela, surtout quand le signe d'affectation n'était pas le signe "=" (vive ALGOL), et à condition de le prononcer de façon imagée : le " := " d'ALGOL lu comme "prend la valeur".

Mais que faire dans le cas d'un appel de procédure ? Que désigne I dans l'appel de procédure TRUC(I) ? ALGOL 60 distinguait l'appel par nom et l'appel par valeur, en faisant directement écho à la façon dont l'appel était compilé : transmission du nom ou de la valeur. C'était une remontée de considérations venues des langages machines, au niveau des langages évolués. En sommes-nous aujourd'hui totalement débarrassés ? Combien de langages font référence à la nature des paramètres : servent-ils de données (auquel cas c'est leur qui importe), ou de résultats (auquel cas, ce qui importe est le nom auquel le résultat sera affecté). Dans cette perspective, une procédure qui admet un paramètre résultat est de la nature d'une affectation, puisque l'on affecte la valeur trouvée à un objet dont on a donné le nom. En conséquence, elle est de la nature d'une instruction. Par contre, une fonction calcule une valeur à partir de la valeur de données, elle ne peut comporter de paramètres résultats, elle est de la nature d'une expression.

Les langages du début des années soixante n'en étaient pas à ce niveau de clarté. ALGOL 60 admettait des paramètres résultats dans les fonctions, et jouait avec les effets indésirables qui en résultent (effets de bord). Ce qui veut dire que les concepts fondamentaux de nom et de valeur, la nature arbitraire du lien qui existe entre un nom et une valeur, n'avaient pas pris encore leur plein déploiement. Mais ils étaient présents dans l'enseignement dès cette période, mal maîtrisés par les enseignants, et, en conséquence, difficiles à communiquer aux étudiants. De la

même façon, le concept d'information était là, avec sa caractéristique que la forme d'une information n'implique pas le sens qu'on peut lui attacher, qu'il dépend d'une interprétation, c'est-à-dire du contexte dans lequel l'information sera lue et prendra une signification.

4. Un corps de doctrine.

Ma venue à l'Institut de Programmation de la Faculté des sciences de Paris n'eut pas pour seul effet de me faire changer la façon de présenter l'ordinateur aux étudiants. Cet institut devait les préparer à des applications variées. Il fallait donc aborder d'autres emplois de l'ordinateur que le calcul scientifique. Il fallait leur parler de fichiers, des supports physiques sur lesquels ils étaient enregistrés, et du fait qu'on ne peut traiter de la même façon un fichier sur bande magnétique (à accès séquentiel) ou sur disques. Il fallait présenter des langages prenant en compte ces fichiers (ALGOL 60 n'était pas en cela un modèle du genre).

On se trouvait ainsi obligé d'évoquer la pluralité des langages, à une époque où ils étaient en pleine prolifération. On tentait de prendre un peu de distance par rapport à eux en expliquant ce qu'ils avaient de commun (fort heureusement, LISP, APL ou SNOBOL, langages aux structures éloignées du lot commun, étaient fort peu répandus et restaient confinés dans d'étroits cercles de chercheurs). L'essentiel du cours sur les langages portait sur leurs grammaires. On parlait aussi de compilation et d'analyse syntaxique.

Les systèmes d'exploitation étaient apparus avec les gros ordinateurs du début des années soixante. On n'en avait pas encore une idée très claire. Le parallélisme de traitements n'était pas encore monnaie courante, les ressources de l'ordinateur n'étaient pas en général partagées entre plusieurs tâches. Mais la gestion de l'ordinateur, avec ses périphériques divers, celle de la bibliothèque de logiciels offerts par le constructeur ou créés dans les centres informatiques, étaient déjà suffisamment complexes pour qu'on en parle aux étudiants.

A côté de cours de calcul numérique, on institua des cours de "traitement de l'information non numérique". On commença à parler de structures de données : chaînes de caractères, files, piles... On fit appel aux chapitres de l'algèbre sur les arbres ou les graphes.

Tout ceci n'était pas toujours très structuré. Mais il y avait là tout un ensemble de cours dont on pensait qu'ils faisaient partie du bagage d'un futur programmeur dont la spécialité d'application était inconnue à l'époque de formation.

J'ai parlé de tout ceci en abandonnant le pronom personnel "je" au profit de l'indéfini "on". C'est que l'expérience que j'ai vécu à l'Institut de programmation de Paris n'était en rien exceptionnelle. L'IFIP, fédération internationale pour le traitement de l'information, avait été créée à Paris en 1959. En 1964, elle se dota d'un "comité technique pour l'enseignement" (TC3) dont je fus nommé membre.

Des collègues universitaires se réunissaient ainsi deux fois par an, sous la présidence du Professeur Buckingham, de Londres. Nous mettions nos expériences en commun. Nous nous étions donné pour tâche de rédiger un programme d'études supérieures en quelque chose qu'il fallait bien nommer. Je n'ai pas souvenir que nous ayons consacré une longue discussion au nom à adopter. Les anglophones utilisaient "Computer Science" les scandinaves "Datalogy". Les discussions se faisant en anglais, je m'en tenais là à "computer science".

Pour ce groupe, l'existence d'un corps de doctrine original fondé par l'apparition des ordinateurs ne faisait aucun doute. Sa situation n'était pas claire. Dans certaines universités américaines, le département de "computer science" était une sous-division de celui des mathématiques. Dans d'autres, il dépendait du département de génie électrique. Il y eut même des universités où ces deux sous-divisions coexistaient, dans une harmonie douteuse. Ce n'était pas la première préoccupation du TC3. Nous voulions élaborer un programme de cours pour l'informatique. Il parût en 1968.

Je m'étais imaginé que j'avais très rapidement pris conscience de l'émergence d'une science nouvelle, baptisée "informatique" en France. C'est à ce propos que j'ai constaté il y a quelques années que notre mémoire peut nous jouer de mauvais tours. J'ai reçu il y a peu d'années la visite de M. Mounier Khun, jeune historien travaillant en relation avec "l'institut d'histoire du temps présent" du CNRS en France. Il me présenta un document de plusieurs pages (ou plus exactement une photocopie du document), me demandant si c'était bien ma signature qui figurait au bout. Je répondis que oui. Il me laissa alors consulter ce document, dont je n'avais gardé aucun souvenir. C'était le compte rendu d'une réunion organisée par CNRS en 1965 sur la question de savoir s'il convenait de créer dans ce CNRS une section d'informatique. Je n'avais pas été invité à cette réunion (je n'étais pas chercheur du CNRS). Jean Delhay, de l'Observatoire de Meudon, y avait été invité. Mais, empêché d'y participer, ou trouvant que j'étais plus à même d'intervenir, il m'avait demandé de la remplacer. A l'ouverture de la séance, on me demanda de servir de secrétaire de séance, ce que j'eus le tort d'accepter. J'étais donc le rédacteur du fameux compte-rendu que me présentait M. Mounier-Khun, je l'avais signé. Pourquoi insister là-dessus ? Parce que le dit compte-rendu ne comporte aucune trace d'une intervention de moi dans les débats. La plupart des participants considéraient que l'ordinateur était un outil technique au service du calcul, et qu'il n'y avait là rien d'autre, du point de vue scientifique, que des mathématiques. Certes, certains chapitres jusqu'alors marginaux prenaient tout-à-coup une grande importance, comme l'algorithmique ou la théorie Rigal fit valoir un autre point de vue, soutenant que ce corps de doctrine en train de s'élaborer autour des ordinateurs était la marque d'une discipline nouvelle. Mais il était encore trop tôt pour pouvoir en donner une définition qui en marque l'originalité. Je ne suis pas intervenu dans le débat.

A force de fouiller dans ma mémoire, j'ai retrouvé une image confuse de la salle de réunion. Je n'ai d'autre souvenir du débat que ce que j'ai lu dans le compte-

rendu. J'essaie de deviner aujourd'hui pourquoi je n'ai rien dit. Parce que, secrétaire de séance, j'étais trop occupé à noter pour pouvoir prendre la parole ? Aujourd'hui cela ne suffirait pas pour m'imposer le silence sur un sujet pareil. Il est plus vraisemblable que j'étais dans une période de doute ou d'interrogation. Je n'étais pas encore clairement conscient de l'existence d'une science informatique, je l'aurais fait savoir. Mais je ne croyais probablement plus qu'il n'y avait là que des chapitres des mathématiques, j'aurais mêlé ma voix aux défenseurs de cette thèse. Mais ceci n'est que reconstruction après coup. Une seule certitude : en 1965, je n'étais pas encore fermement convaincu de l'existence d'une science informatique.

D'autres faits m'ancrent dans cette idée. Je n'ai jamais cessé de faire de la recherche scientifique, même au milieu de tâches d'enseignement très lourdes, comme c'était le cas à cette époque où il fallait tout créer. Quand j'étais à l'Observatoire de Meudon, j'avais constaté que les chercheurs manquaient de bonnes méthodes numériques pour le calcul d'une intégrale. Je m'étais donc lancé dans l'étude de l'approximation des opérateurs linéaires. Pour approcher l'intégrale d'une fonction sur un certain intervalle, on calculait exactement l'intégrale d'un polynôme approchant la fonction sur cet intervalle. Au lieu de cela, je me demandais comment calculer exactement un opérateur linéaire approchant celui d'intégration, appliqué à la fonction elle-même. J'ai été invité comme conférencier au colloque de l'IFIP à New York en 1965 : c'est de cela qu'on m'avait demandé de parler. En 1965, j'enseignais les ordinateurs, je faisais des recherches en analyse numérique.

5. Une science nouvelle.

Je suis incapable de dire quand je fus convaincu de l'existence d'une science nouvelle. J'ai un autre point de repère très précis dans ma mémoire, en fin avril début mai 1968. J'avais été invité par IBM à un séminaire international à Blaricum, en Hollande. C'était un séminaire polyglotte, avec traduction simultanée. Nous avions donc des écouteurs, avec de petits boîtiers munis d'un bouton d'un plusieurs positions, pour le français, l'anglais... Une affiche disait à quoi correspondaient les différentes positions du bouton, sauf la dernière. C'est le genre de choses qui intrigue un physicien, je n'eus donc rien de plus pressé que d'essayer la dite position. J'entendis une émission de radio en langue française. C'était très commode: quand l'exposé devenait trop technique, ou sortait de mes propres préoccupations, j'écoutais la radio, l'air très attentif à ce que disait le conférencier. C'est ainsi que j'appris qu'il y avait de fortes manifestations étudiantes à Paris, que la police bloquait l'accès à la Sorbonne. Aucun doute possible : c'était le début de mai 68, car je fus totalement surpris : il n'y avait rien en France quand j'étais parti à Blaricum.

Un autre souvenir amusant de ce séminaire. Un conférencier vint présenter une expérience ambitieuse d'enseignement assisté par ordinateur qu'il était en train de développer. Un participant lui parla de machine de Turing. Il répondit qu'il n'avait pas consulté ce constructeur. Ce sont des petits détails de ce genre qui aident à garder trace des événements !

Il est très vraisemblable que nous avons discuté dans ce séminaire de la nature de l'informatique : science nouvelle, ou simple branche des mathématiques. Là encore, je n'ai pas de souvenir précis d'une séance organisée sur ce thème, avec exposé et discussion. Je pense même qu'une telle séance n'eut pas lieu, elle m'aurait probablement marqué. La discussion a dû être informelle, entre participants intéressés par question, hors séance. C'est au cours de ce séminaire que Perlis, brillant chercheur et enseignant en ces années-là, me remit un texte photocopié, que j'ai gardé, dans lequel il discutait de la nature de la "computer science". On peut faire valoir que les américains ont été les premiers à percevoir l'émergence d'une science nouvelle, car le terme de "computer science", qui comporte le mot "science", date d'environ 1965. Mon collègue M. P. Shutzenberger, qui connaît bien les américains pour avoir travaillé plusieurs années dans des universités de ce pays, notamment avec Noam Chomsky, dit que le mot ne prouve rien. Les américains, selon lui, ne sont pas très attentifs aux mots qu'ils utilisent, on met "science" sur sa carte professionnelle parce que cela va de pair avec "université". Ce n'est en rien une prise de position épistémologique.

Perlis avait pris au sérieux le terme de "science", et se demandait ce qu'elle était. Son texte répondait à la question en creux, en éliminant ce qu'elle n'est pas. Elle n'est pas la science des ordinateurs, malgré son nom ("computer science"), parce qu'il n'y a pas de science d'un instrument : l'astronomie n'est pas la science des télescopes, ni la thermodynamique celle des thermomètres. Elles n'est pas, à la façon des sciences naturelles, une science d'observation qui étudierait les ordinateurs pour en découvrir les lois ou les classer en genres et en espèces : nous n'avons pas à étudier ces machines que nous avons conçues et fabriquées. Elle n'est pas la science du calcul ("computing science", comme certains ont proposé de l'appeler), parce que la science du calcul existe depuis longtemps, et s'appelle la mathématique. Il restait à dire ce qu'elle est.

J'ai gardé un souvenir fiable d'interminables réunions comme nous avons du en subir en mai et juin 1968, pendant lesquelles je n'écoutais que d'une oreille (ce n'était pas toujours très captivant). Sans aller jusqu'à mettre en pratique ce que nous interdisait notre professeur de physique qu'en j'étais en seconde ("Ah, écoutez, vous! vous ne pouvez pas écouter d'une oreille et lire de l'autre"), je pensais qu'il serait bon d'écrire un livre d'informatique pour un grand public, j'en construisais le plan. Je commençais à l'écrire au dernier trimestre de 1968. Ressortit alors la question de l'information, que je m'étais bien gardé d'aborder avec les étudiants. Qu'est-elle, pour que sa forme ne suffise pas à dire sa signification, et que même celle-ci soit à ce point arbitraire que je peux en prendre le sinus ou m'en servir comme instruction pour l'ordinateur (voir plus haut) ? Puis, un jour de 1969, pendant que j'étais en vacances en Haute-Loire, je me suis demandé si la dualité nom-valeur s'appliquait aussi à l'information, et si oui, était-elle de la nature d'un nom ou d'une valeur, d'une adresse ou d'un contenu ? La réponse vint aussitôt : elle est un contenant, dont la signification est le contenu. L'information est comme une boîte, dont l'aspect ne suffit pas à dire ce qu'il y a dedans. J'ai cru pendant longtemps que c'était une particularité de ce qu'on met dans des cases de mémoire,

et qu'on a peut-être abusivement nommé "information". Puis un jour, dans un château de l'Orne que fréquenta la Marquise de Sévigné, j'ai lu sur un cadran solaire "TIME".

Evident, l'heure en anglais. Malheureusement, il y avait devant "ULTIMAM". "Ultimam time". Pourquoi ce mélange de latin et d'anglais ? Il m'a fallu longtemps (peut-être une demi-minute), pour découvrir en "TIME" l'impératif du verbe latin "timeo" : "crains la dernière". La donnée des quatre lettres "t i m e" ne suffit pas à déterminer la signification du mot, il y faut autre chose venu de l'extérieur.

A partir de là, pour moi, tout se mettait en place. L'information est un texte, une suite de lettres. Ce n'est pas ce qu'y voit le journaliste ou l'historien, ce qui permet de distinguer l'informatique de l'histoire ou du journalisme : tous se préoccupent d'information, l'informatique seule s'en tient aux lettres qui la composent.

Admettons. Comment relier cette information à celle du sens commun ? Pour celui-ci, l'information est ce qui apporte une connaissance. Si elle est écrite, elle a bien l'aspect d'un texte. Mais je ne peux pas lire un texte en japonais, je n'apprends rien d'un texte de mathématiques si je n'y comprends rien, je n'apprends rien d'un livre élémentaire car je sais déjà ce qu'il dit. L'apport de connaissance est subjectif, il n'y a de science que du général, seul le texte a un caractère objectif. Tout s'enchaînait fort bien: l'informatique est la science du traitement de l'information considérée comme le support formel des connaissances. Par un hasard providentiel, c'est à très peu près la définition qu'adoptait l'Académie Française, proposée par le même Philippe Dreyfus qui avait créé le mot "informatique". Il m'a dit regretter cette définition : pour moi, elle relevait du génie !

Ceci posé, la démarche informatique est évidente. On part de ce que l'on sait, disons de "connaissances données". On les représente avec des lettres ou des chiffres sous forme "d'informations données". On traite ces informations d'après leur seule forme et suivant les règles de la grammaire qui a présidé à leur écriture, traitement formel, puisque basé sur la seule forme. Le résultat est un texte formel. On l'interprète dans le contexte où on sait qu'il se situe (parce que celui qui a représenté et organisé le traitement connaît ce contexte). J'ai longtemps insisté sur le problème de la représentation des connaissances données : elle doit être choisie telle que la forme obtenue implique ces connaissances, et ne puisse être interprétée autrement, faute de quoi on ne saura pas à laquelle des interprétations possibles rattacher le résultat. J'ai aussi beaucoup insisté sur le problème du traitement, car l'idée qu'on puisse traiter un texte sans s'occuper de ce qu'il peut vouloir dire était très neuve (du moins le croyais-je à l'époque. J'ai découvert depuis que dès le XVII^e siècle, le philosophe anglais Thomas Hobbes avait demandé que l'on calculât avec les mots pris indépendamment de leur signification comme on calcule avec les nombres). J'ai découvert en mai 1995 que l'interprétation des informations résultats pouvait elle aussi être source de difficulté, principalement parce qu'on oublie le

contexte dans lequel il convient de les lire. Lors de l'élection présidentielle en France, nous avons été abreuvés de sondages d'opinion, auxquels les résultats du vote ont paru donner tort. On a dit que les sondages s'étaient trompés, qu'ils avaient été mal faits, voire qu'ils étaient mauvais dans leur principe. On avait tout simplement oublié de tenir compte d'un résultat pourtant fourni par ces sondages : le nombre d'indécis. Si l'on s'était posé la question de savoir comment pouvaient voter ces indécis, on obtenait des résultats très différents suivant qu'ils penchaient pour l'un ou l'autre : les résultats effectifs se sont sagement rangés dans ces fourchettes d'indétermination.

Ainsi, pour moi, tout était clair. J'avais un schéma cohérent de l'informatique, qui rendait compte des phénomènes réellement observés dans les applications de cette science. Il faisait apparaître des points sensibles correspondant à des difficultés concrètes. Il expliquait pourquoi la logique formelle est un des supports théoriques de cette science. Il répondait à une autre de mes préoccupations. On avait baptisé les ordinateurs "cerveaux électronique", et, pour moi, c'était inacceptable : je ne crois pas être de la nature d'une machine (c'est une croyance fortement ancrée en moi, et si je ne cherche à l'imposer à personne, nul ne peut me l'interdire). La différence fondamentale entre la machine et moi était là : pendant qu'elle traite des caractères, je réfléchis à des idées.

6. Aujourd'hui : le jugement n'est pas encore rendu,

Avec l'aimable autorisation de la rédaction, nous reproduisons ici un article paru dans "Le Canard Enchaîné", en date de l'événement :

"Un curieux procès vient de se dérouler dans la 12^{ème} chambre correctionnelle de Paris. Rappelons brièvement les faits. En 1979, J. C. Simon remit au Président de la République un rapport où il recommandait un enseignement obligatoire d'informatique à tous les lycéens. En 1980, le Ministre de l'Education Nationale, M. Beulac, demanda au professeur Jacques ARSAC (J. A. dans la suite) de créer un enseignement optionnel d'informatique dans les lycées. Partie de 12 Lycées, l'expérience se répandit pour atteindre 600 lycées, soit plus de 60 000 élèves, en 1990. Cette année là, à la suite d'un autre rapport d'un groupe d'experts, l'option fut supprimée par le nouveau ministre, M. Jospin. J. A. écrivit alors plusieurs articles très virulents, mettant en cause les experts. Ceux-ci portèrent plainte, et l'affaire est venue en jugement. Les experts étaient représentés par leur avocat, Maître Bafouillet (M. B. dans la suite). J. A. a décidé de se défendre lui-même. Très rapidement, les débats ont fait apparaître que toute la question repose sur la nature de l'informatique. Si elle est une technologie nouvelle, son utilisation repose sur un savoir-faire, qu'on acquiert dans des ateliers de pratique, comme ceux mis en place par M. Jospin. Si au contraire elle est une science, elle relève du savoir, fait partie de la culture et doit s'enseigner comme une discipline scientifique. J. A. a soutenu que l'informatique est une science, celle du traitement de l'information considérée comme le support formel des connaissances. Il a prétendu que pour fournir une connaissance nouvelle, on commence, en informatique, par représenter les données par des suites de caractères suivant la

grammaire d'un langage, puis qu'on traite ces suites par des règles judicieusement choisies dans la grammaire, et qu'enfin on interprète les suites de caractères résultats en connaissances cherchées. Une vive discussion s'en est suivie entre Maître Bafouillet et J. Arsac, que nous résumons ici.

M. B. Votre thèse est une absurdité. Vous dites qu'on représente les connaissances données par des suites de caractères dont on ne considérera que les lettres, sans faire intervenir leur sens. C'est manifestement impossible. Or l'informatique est très largement utilisée aujourd'hui.

J. A. Il n'est pas difficile de montrer par des exemples qu'il y a des domaines où on peut ne considérer que la forme des données, pas leur sens. C'est par exemple le cas dans un vote : le dépouillement ne fait pas intervenir le parti des candidats, ni les idées qu'ils défendent. Pourtant, personne n'en conteste la validité.

M. B. Vous avez trouvé un exemple où votre schéma fonctionne. Mais la forme dans laquelle sont codées les données n'est qu'un aspect de celles-ci. La non considération de leur sens est un moins. Vous ne ferez pas sortir le plus du moins. Ce qui est perdu à l'entrée l'est définitivement.

J. A. Je vois que Maître Bafouillet a lu le livre "Apport de l'informatique à l'enseignement des mathématiques", que J. Kuntzman a publié chez CEDIC en 1974. Mais cela ne contredit pas ma thèse, bien au contraire, parce que c'est ce que l'on observe. Le correcteur orthographique d'un traitement de texte peut corriger les fautes de forme (éléfan), pas celles liées au sens (le mâle s'enfuit impuissant, au lieu de le mal...).

M. B. M. Arsac est un brillant sophiste, qui a réponse à tout. Mais il ne peut rien prouver avec des exemples, c'est un fait bien connu en sciences.

J. A. Je le reconnais volontiers. J'ai construit ma thèse en considérant de nombreux exemples. Mais j'ai procédé à une généralisation, en posant comme un principe ce que j'ai constaté dans tous ces exemples. Carnot n'a rien fait d'autre quand il a posé son principe de la thermodynamique, que nul ne conteste aujourd'hui.

M. B. Voilà que J. Arsac ose se comparer à Carnot ! Fort heureusement, il y a bien d'autres raisons de refuser sa thèse. Le schéma qu'il prétend être typique de l'informatique est en fait celui de la communication. Je représente les idées que j'ai en tête par des lettres sur une feuille de papier. Le lecteur les déchiffre, et les interprète pour y retrouver mes idées.

J. A. Objection aisément réfutée. Dans la communication, il n'y a pas traitement : les lettres que j'ai écrites sont lues, et non l'objet de calculs tellement complexes qu'ils sont généralement infaisables sans l'aide d'une machine.

M. B. De toutes façons, le prétendu schéma de l'informatique est inacceptable, et ne peut qu'être le fruit d'un esprit atteint de paranoïa. C'est en effet le schéma de toutes les sciences. Le physicien observe des phénomènes, les représente par des théories formelles, fait des calculs mathématiques qui ne dépendent pas de la signification physique des éléments du calcul, puis interprète les résultats formels en terme de phénomènes physiques.

J. A. Je félicite Maître Bafouillet pour sa connaissance très profonde de la physique. Mais l'activité d'observation des phénomènes, le choix d'un bon support théorique sont tout autre chose que ce que fait l'informatique. Après avoir remarqué qu'une baguette de verre frottée avec une peau de chat attire des petits bouts de papier, poser comme modèle formel qu'on a arraché des particules électrisées à la baguette de verre est d'une autre nature que représenter des sonnées par des caractères (comme par exemple célibataire par 1, marié par 2, veuf par 3...) Il est vrai qu'une fois le modèle théorique élaboré, il est traité formellement ce qui se fait aussi en informatique. L'informatique a pris aujourd'hui une place de choix dans la simulation de phénomènes physiques. Ainsi mon schéma n'englobe pas celui de la physique, mais, lorsque les schémas coïncident, ou plutôt dans la partie des opérations physiques où ils coïncident, alors l'informatique est une aide puissante pour la physique.

M. B. Je pourrais aisément accuser J. Arzac de plagiat. En 1962, Péter Lucas, dans sa thèse, a montré que l'ordinateur est de la famille des machines de Turing, qu'il ne peut faire que des traitements formels, et donc qu'il ne peut rien pour les propriétés indécidables.

J. A. Exact. Mais il n'y a pas plagiat. Lucas n'a parlé que des ordinateurs. Il n'a rien dit de la façon dont on lui fournit des données formelles, ni du problème d'interprétation du résultat. En outre, il n'a rien dit de la possibilité de traitements formels sans ordinateurs, ce qui rend impossible toute généralité. On rejette ainsi des traitements de nature informatique apparus bien avant les ordinateurs (comme le vote), et on se place dans une situation très délicate si un jour on invente une autre machine à traiter l'information qui ne soit pas du type des ordinateurs. Je signale à Maître Bafouillet que, dans l'exposé qu'il fit quand on lui remit la "Turing Award" de l'ACM (association for computing machinery), Herbert Simon définit la science des ordinateurs (mot anglais équivalent au français "informatique") comme "l'étude des phénomènes entourant les ordinateurs". Il la présente comme une science expérimentale, dont le principal but est la description d'un modèle de fonctionnement de l'esprit humain, la mettant ainsi au coeur des sciences cognitives. Cet exposé date de 1975. Sa définition de l'informatique n'est pas identique à la mienne, mais il y a là aussi une claire affirmation de ce qu'elle est une science.

M. B. Le professeur Arzac abuse de la supériorité que lui donne un long passé d'enseignement à l'Université. Mais il n'a qu'un savoir académique. L'important de l'informatique n'est pas ce qui s'enseigne dans des maîtrises

spécialisées qui ne touchent qu'un petit nombre d'étudiants, c'est ce qui pratique quotidiennement dans un grand nombre d'entreprises comme les banques ou sociétés d'assurance, ou dans de toutes petites entreprises, chez le plombier ou la garagiste qui font leur gestion sur ordinateur ou tapent leur courrier avec un traitement de texte. Ceux-là n'ont fait aucune étude, ne savent rien d'une hypothétique science. Pourtant, ils font de l'informatique tous les jours.

J. A. Je pense que, sur sa lancée, Maître Bafouillet va nous expliquer que le marchand de légumes, au marché, qui calcule le prix des deux salades avec sa calculette de poche, fait des mathématiques. Pourtant, il n'a fait aucune étude sérieuse de cette hypothétique science donc les mathématiques ne sont pas une science.

Le débat s'est poursuivi un certain temps de cette façon là, J. A. réfutant l'une après l'autre les diverses objections élevées par la partie civile. S'agissant des accusations portées par J. A. contre les experts, on en est resté à des procès d'intentions. Les accusations portées par les experts contre l'enseignement de l'option furent réfutées, mais toujours dans des cas particuliers. Pouvait-il en être autrement ? Celles portées par J. A. contre les experts furent déclarées relever d'erreurs de lecture, ce n'était pas ce que les experts avaient voulu dire. Après un long débat, M.B. déclara dans son réquisitoire que J. A. était habile à présenter ses idées et à leur donner l'apparence de la vérité, que cela pouvait troubler des non initiés, comme lui-même, mais que la communauté internationale ne s'y était pas trompée. Il y a plus de 25 ans que J. A. a présenté sa thèse sur l'informatique comme science, et pourtant elle n'a pas encore été adoptée par la communauté internationale. L'Académie des sciences, en France, n'a pas encore pris clairement parti sur cette question, et le mot "informatique" ne fait partie de l'intitulé d'aucune de ses sections. Dans sa réponse, J. A. nota que ce genre d'institution n'évolue que lentement, que la définition du statut de l'informatique n'est pas seulement l'affaire de cette discipline, mais à part égale de la philosophie des sciences. Qu'il faut donc que des informaticiens se lancent dans la philosophie, ou que des philosophes s'initient à l'informatique, et que tout cela prend du temps. Pendant que se déroulent ces discussions, l'informatisation de la société progresse. Les élèves ne recevant aucun enseignement de la science informatique ne voient que des ordinateurs, et ne comprennent rien à ce qui est véritablement en jeu. Tel Cassandre, il a prédit les pires catastrophes si on continue à informatiser une société d'analphabètes. Mais qui a jamais écouté Cassandre ? Le Président a marqué un grand embarras à la fin des débats. Certes, dit-il, J. A. a manifestement insulté les experts du Ministre de l'Education Nationale, mais ceux-ci avaient fait preuve de beaucoup de légèreté dans la rédaction de leur rapport. Quant à savoir si l'informatique est une science, les arguments présentés par J. A. paraissent très convaincants à un non spécialiste. Mais l'absence de réponse ferme de la communauté scientifique internationale incite à la prudence. Il a donc renvoyé le jugement à plus tard, sans donner de date".

Le lecteur me pardonnera cette fiction. Elle avait l'avantage de permettre une présentation rapide et immodeste du débat actuel. Sa conclusion est la seule qui puisse être donnée aujourd'hui. Le débat n'est pas définitivement tranché. Je ne peux donner que ma grande conviction que sera finalement reconnue la vraie nature de l'informatique, science du traitement de l'information considérée comme le support formel des connaissances.

Jacques ARSAC

Professeur émérite à l'Université P. et M. Curie
Correspondant de l'Académie des Sciences

La situation en Tunisie

Les Orientations Nationales de l'Informatique Scolaire en Tunisie

Au niveau du VIIIème Plan 1992-93 à 1996-97, les ressources humaines ont constitué une préoccupation majeure qui conditionne le développement du secteur dans son ensemble.

Les objectifs fixés sont :

- La Consolidation de l'appareil de formation par l'augmentation de la capacité d'accueil des institutions de formation, le renforcement du corps enseignant, le lancement d'une formation doctorale, la mise en place des cycles de formation continue, la reconversion à l'informatique;
- La Création de départements informatiques dans toutes les écoles d'ingénieurs et tous les instituts de gestion ;
- La Généralisation de modules d'initiation dans tous les cursus d'enseignement secondaire et supérieur au niveau du premier cycle ;
- L'Organisation du secteur des écoles privées (homologation des diplômes, instauration d'un examen national de référence, cahier de charges référentiel...).

Dans ce qui suit on présentera la stratégie nationale pour l'expérience tunisienne dans l'introduction de l'informatique dans l'enseignement secondaire et ceci en tant que discipline d'enseignement mais aussi en tant qu'outil de rénovation pédagogique.

Au niveau de l'enseignement primaire, il est prévu de lancer une série d'expérience d'applications pédagogiques dans les différentes disciplines.

Au niveau de l'Enseignement Secondaire :

L'introduction de l'informatique à l'école tunisienne a démarré en 1983-84 par l'équipement de trois lycées par des laboratoires de micro-ordinateurs (lycée technique de Radès, collège Sadiki et lycée de garçons de Sfax). Avec la création du CBMI en 1985, devenu en 1990 INBMI, cette action a touché à titre expérimental et dans sa première phase 2 lycées pilotes et cinq lycées témoins par un enseignement de l'informatique comme outil de travail à partir de la première année d'étude. Parallèlement à cet enseignement, les expériences d'applications pédagogiques de l'ordinateur se déroulaient à partir de didacticiels et selon le modèle de l'EAO.

Depuis l'année scolaire 1990-91 et dans le cadre de la réforme du système éducatif, l'introduction de l'informatique scolaire s'est concrétisée par un scénario de mise en oeuvre qui vise dans sa première phase, à permettre à tout élève qui achèverait le secondaire de maîtriser l'outil informatique et dans une deuxième

phase des objectifs plus ambitieux relatifs à l'acquisition d'habilités cognitives et à la rénovation pédagogique.

Cette action a touché l'année scolaire en cours plus de 34 436 élèves et permettrait de toucher à la dernière année du plan (Septembre 1996) un effectif élèves d'environ 46000.

Les objectifs attendus :

L'action d'introduction de l'informatique scolaire vise à moyen terme, à permettre à tout élève qui achèverait le secondaire de maîtriser l'outil informatique. Pour ce, un scénario progressif a été mis au point et permettra d'atteindre à partir de septembre 1999 les objectifs visés.

L'enseignement de l'informatique est dispensé aux élèves des 3ème et 4ème (6ème et 7ème ancien régime) en tant que matière à option. Progressivement, la discipline informatique se généraliserait pour toucher tous les élèves de 3ème année des sections Economiques, Techniques, Maths et Sciences Expérimentales au cours de l'année scolaire 1999/2000 et ceux de la section Lettres à la fin du IXème plan et pourrait devenir alors obligatoire en 3ème année tout en restant optionnelle en 4ème année.

Parallèlement à cet enseignement, les élèves, les enseignants et les pédagogues seront initiés aux différents usages de l'informatique dans des applications pédagogiques dans les différentes disciplines à fin d'oeuvrer pour la pratique d'un nouveau type d'apprentissage longtemps souhaité par les réformes éducatives.

Bilan des réalisations :

A- Les effectifs élèves touchés par la matière informatique :

L'introduction de la matière informatique a permis de toucher en 1992/93 environ 9000 élèves (en 6ème et 7ème), 17000 élèves en 1993/94 et 25000 élèves en 1994/95.

Pendant l'année scolaire 1995-96, environ 34500 élèves suivent les cours de la matière informatique à raison de 2 heures par semaine. Ce nombre est réparti comme suit : 19000 élèves en 6è année, 15500 en 7è année. En plus 3148 élèves suivent régulièrement le cours d'informatique dans les six (6) Lycées Pilotes.

Les taux de couverture correspondants sont de l'ordre de 62%, 37% et 4% respectivement en 6ème année secondaire des sections Economie et Techniques, Maths et Sciences Expérimentales et Lettres, et de 59%, 27% et 2% en 7ème année secondaire des mêmes sections.

En ce qui concerne la section Lettres, les programmes actuels ne sont pas adaptés aux spécificités de la section et continuent donc à se faire à titre expérimental dans quelques établissements à raison de 4% des effectifs.

B- Les ressources humaines :

1- Les enseignants :

L'enseignement de l'informatique est assuré par des enseignants ayant une formation universitaire variée : des maîtres en informatique et des professeurs d'autres disciplines en exercice.

Durant l'année scolaire en cours, cet enseignement est assuré par 288 enseignants dont 221 à plein temps. Ils sont répartis en 110 maîtres en informatique, 178 provenant des autres disciplines (111 chargés à plein temps et 67 à temps partiel). Ces derniers ont généralement suivi des stages de formation assurés par l'INBMI .

Il y a lieu de remarquer que la coexistence de ces deux profils d'enseignants offrirait de meilleures garanties à la discipline informatique et à la réalisation de ses objectifs.

En effet, la présence des maîtres en informatique est de nature à élever le niveau scientifique des enseignants et ceci par émulation. Quant à la présence des enseignants des autres matières, qui sont bien motivés et bien préparés pédagogiquement, elle est de nature à faciliter l'ouverture de la discipline informatique aux autres disciplines et à mieux favoriser les applications pédagogiques de l'ordinateur dans les différentes disciplines.

Au niveau du profil de ces enseignants, les compétences de base nécessaires aux formateurs de cette discipline ont été définies en deux temps :

- à court terme pour enseigner le programme actuel;
- à moyen et à long terme pour enseigner un programme révisé et élargi comprenant un volet plus étoffé de culture informatique et d'activités plus adéquates.

Le programme de formation continue à ces formateurs prévoit :

- un complément de formation en culture générale sur les nouvelles technologies d'information et leurs impacts dans les sphères socio- économiques;
- une formation sur la didactique de l'informatique;
- un complément de formation en Science Informatique (pour les enseignants non spécialistes);
- une formation sur les nouvelles technologies d'information éducationnelles.

2- Les encadreurs pédagogiques :

Actuellement, un inspecteur et cinq conseillers pédagogiques exercent à travers tout le pays et assurent l'encadrement pédagogique, le suivi et la formation de tous les enseignants de la matière. Ils organisent les leçons modèles, les journées pédagogiques et les journées de formation à leur profit. Ils rendent compte également à l'INBMI de leurs remarques et observations pour les besoins de suivi. De plus, ils participent à l'élaboration de manuels de cours et d'exercices.

3- Les opérateurs de laboratoires :

Il a été constaté, ces dernières années, que l'accès aux laboratoires était lié à la présence de l'enseignant, ce qui a limité les activités dans les laboratoires et a empêché leur utilisation par les enseignants des autres disciplines. De plus, des difficultés ont été rencontrées dans la tenue du matériel en place et surtout au niveau des logiciels. Afin de garantir une utilisation optimale des laboratoires, il est envisagé de doter progressivement chaque lycée d'un opérateur de laboratoire, particulièrement pour les lycées qui seront équipés d'un deuxième laboratoire.

C/ Le programme d'enseignement de l'informatique :

Dans le cadre de l'élaboration des nouveaux programmes d'enseignement et après concertation avec des experts internationaux, les objectifs retenus par l'enseignement de l'informatique à partir de septembre 1993 ont été fixés par le Décret n° 93-670 du 29 Mars 1993 Art 635....Art 638. Ces objectifs visent à :

- Initier les élèves à la culture informatique en donnant de l'informatique une image dynamique qui en reflète les usages multiples dans la société et leur permettre de suivre son évolution pour mieux la comprendre et la maîtriser;
- Développer chez les élèves l'habilité à résoudre des problèmes selon des démarches rigoureuses et systématiques, en faisant appel à des outils de travail et de développement. Les problèmes traités se rapporteront dans la plupart des cas aux différentes disciplines d'enseignement et ceci pour une meilleure osmose interdisciplinaire et afin de développer progressivement chez les élèves et les enseignants la motivation pour s'appropriier l'ordinateur et s'en servir comme un instrument d'apprentissage;
- Préparer les enseignants et les élèves à un futur usage de l'ordinateur comme outil d'apprentissage.

D/ Les laboratoires informatiques :

La concrétisation de la discipline informatique s'est effectuée par l'implantation de 109 laboratoires informatiques en 1991/92 dans le cadre du programme de la coopération Tuniso-Italienne. Durant l'année scolaire en cours, le nombre de laboratoires est élevé à 160. Ces laboratoires informatiques comportent chacun (9+1) ou parfois (7+1) postes avec deux imprimantes partageables.

Les perspectives :

Un scénario de mise en oeuvre de l'action de généralisation de l'informatique a été élaboré et permettra d'atteindre cet objectif pour les sections Economie/Technique en septembre 1998, pour les sections Maths et Sciences Expérimentales en septembre 1999 et la section Lettres en septembre 2001.

Cette généralisation ne concernera que les élèves des 3ème années du secondaire. En 4ème année, la matière restera optionnelle.

Cette généralisation permettra de toucher au total en septembre 1998 environ 81000 élèves répartis en 100%, 80% et 35% pour les 3ème années des

sections Economie/Technique, Maths et Sciences Expérimentales et Lettres et 45%, 35% et 5% pour les élèves de 4ème années des mêmes sections.

En septembre 2001, l'année de sa généralisation totale, environ 108000 élèves enregistrant 100% pour les 3ème années des sections Economie/Technique, Maths et Sciences Expérimentales et Lettres, et 45%, 40% et 10% pour les élèves de 4ème années des mêmes sections.

Pour faire face à cette généralisation, le programme des ressources humaines et matérielles nécessaires a été élaboré comme suit :

A- Les laboratoires informatiques :

L'étude de ce scénario a permis d'élaborer un projet d'implantation des nouveaux équipements dans les lycées. Ce projet tient compte du renouvellement du parc du matériel en place avec un délai estimatif de rotation de 6 années. Compte tenu des effectifs des élèves dans le lycée, des deuxièmes laboratoires seront à prévoir.¹

Les besoins estimés devraient élever le nombre total de laboratoires à 203 en septembre 1996 soit 43 nouveaux laboratoires et 34 laboratoires de renouvellement. Ainsi le nombre total de micro-ordinateurs nécessaires sera de 770 et le nombre de laboratoires à aménager de 43.

Les investissements nécessaires pour la dernière période du VIIIème plan s'élèveront à 1.540.000 DT pour l'acquisition du matériel informatique et 172.000 DT pour l'aménagement.

Durant le IXème plan, le nombre total de nouveaux laboratoires à créer est estimé à 278 et le nombre de laboratoires à renouveler est de 340. Le nombre total de micro-ordinateurs nécessaires à acquérir sera de 6180 et le nombre de laboratoires à aménager sera de 278.

Les investissements nécessaires durant le IXème plan s'élèveront à 12.360.000 DT pour l'acquisition du matériel informatique et 1.112.000 DT pour l'aménagement.

B- Les enseignants :

Les besoins estimés en enseignants devraient élever le nombre total des professeurs d'informatique à 230 en septembre 1995 et à 304 en septembre 1997.

Durant le IXème plan le nombre total d'enseignants est estimé à 436 en septembre 1997 et atteindra 720 enseignants en septembre 2001.

Le déficit en heures d'enseignement sera comblé par le recours aux heures complémentaires ou supplémentaires des enseignants des autres disciplines.

C- Les applications pédagogiques de l'ordinateur :

1- Les applications développées :

Depuis quelques années, certaines applications pédagogiques de l'ordinateur, bien qu'expérimentales dans certaines matières de l'enseignement secondaire en Tunisie, se développent régulièrement et sont exploitées en classe sous la supervision (totale ou partielle du cadre inspectoral et pédagogique). D'autres applications sont en cours de réalisation à l'INBMI par des équipes pluridisciplinaires d'enseignants et de développeurs en informatique. Cette action a été précédée par des études relatives à la conception de l'environnement d'apprentissage basées sur les objectifs visés par les programmes officiels d'enseignement et tenant compte de la réalité de l'environnement éducatif tunisien.

Il existe également des applications réalisées par les élèves de certains lycées. cf. en annexes certaines applications:

- EDIPO de français (le monde merveilleux des inventions et découvertes)
- EDIPO de mathématique
- Visite Guidée au Musée National du BARDO (application multimédia)
- Le XVI^è siècle en France (base de données documentaires)
- Le livre de la rhétorique arabe (Kiteb El Balagha)

2- L'intégration de l'ordinateur dans l'enseignement technique :

Dans le cadre de la réforme du système éducatif, les disciplines d'enseignement technique ont été conçues officiellement selon une approche systémique beaucoup plus que technologique. Cette approche est basée sur la conception assistée par ordinateur, sur la modélisation et la simulation des procédés de fabrication à l'aide de la robotique et de l'ordinaire.

Tous les élèves des classes de technique utilisent donc, dans les programmes officiels, les outils informatiques de conception tel qu'Autocad et Orcad et les outils de fabrication assistée par ordinateur.

Le nombre de poste de travail est estimé à l'heure actuelle à 2000 postes munis de logiciels de conception et d'accessoires d'ordinaire (mini tour à commande numérique, robots, bras automatique,.....)

D- information :

Il est important de faire connaître dans tout le milieu éducatif, les nouveaux développements dans le domaine de l'informatique scolaire et de faire connaître également les réalisations des professeurs et de les mettre en valeur. A cet effet, l'INBMI publie depuis mars 93 un bulletin de liaison trimestriel et le diffuse à un très large public.

E- La formation socio-professionnelle :

En ce qui concerne la formation à la Bureautique, au profit des fonctionnaires des secteurs public et privé, en plus du personnel du Ministère de l'Education, environ 1200 ont bénéficié de cette formation par le biais de

conventions établies avec leurs organismes (Jeunesse et Enfance, Propriété de l'Etat, Conservation Foncière, Banque Centrale et Régie de Tabac).

Au niveau du Ministère de l'Education, la formation a concerné les 128 fonctionnaires des lycées secondaires pour l'utilisation de l'application SYGEL, les 64 professeurs d'économie en surnombre pour leur donner une formation double compétence ainsi que 540 professeurs en exercice motivés par l'informatique.

F- La maintenance des équipements et des logiciels :

L'expérience de l'INBMI avec ses 3 centres régionaux (Sousse, Sfax, et Gafsa) dans la maintenance préventive et curative de tous les équipements des lycées a permis de faire face à la multitude très variée du matériel en place. Chaque laboratoire est visité deux fois par an par l'équipe de maintenance qui procède d'abord à l'inventaire du matériel en place et aux interventions préventives et curatives ensuite. Cependant les difficultés rencontrées actuellement au niveau des moyens de déplacement des techniciens, rendent cette action de plus en plus difficile. Ces difficultés s'accroîtront au fur et à mesure que le parc s'élargit et vieillit, d'où la nécessité d'y remédier dans les meilleurs délais.

Depuis sa création, l'Unité de Maintenance a fourni des efforts considérables pour la réparation du matériel informatique. Les deux actions principales qui ont été menées touchent :

- L'amélioration de la maintenance préventive afin de diminuer le nombre de pannes. En début de semestre, un planning de visite systématique est établi mais reste cependant subordonné à la disponibilité des moyens de transport;
- La maintenance curative par les moyens propres de l'INBMI.

G- Les applications de gestion :

Dans le cadre de l'informatisation de l'administration scolaire, le Système de Gestion scolaire informatisé (SYGEL) a touché à l'heure actuelle 380 lycées secondaires et a concerné des opérations nouvelles comme la clôture de l'année scolaire, l'orientation scolaire et certaines statistiques.

De plus et avec la collaboration du CNI, un outil relatif à la gestion budgétaire "IGEBI" a été testé par certains lycées secondaires et sera généralisé progressivement.



La situation en Belgique

Charles DUCHATEAU

Le contexte général

· L'enseignement en Belgique est du ressort des 3 communautés qui composent l'état fédéral: communauté française, communauté flamande et communauté germanophone.

· La nécessité de restrictions budgétaires en communauté française a conduit à revoir les normes d'encadrement dans les écoles. Ces restrictions devraient conduire, d'une part à une perte d'environ 10% des emplois dans l'enseignement, et d'autre part à un recentrage des curriculums sur les disciplines de base (français, mathématique, langues) au détriment des options dont la plupart sont appelées à disparaître.

· Dans ce contexte nouveau, chaque école aura davantage d'autonomie pour définir un "projet d'école" sur lequel porter ses efforts. Certaines écoles pourraient donc continuer à promouvoir l'enseignement et les utilisations des NTIC.

· On peut cependant s'attendre à voir disparaître l'option informatique qui existait dans certaines écoles au niveau de la fin du secondaire, et intitulé "Education par la technologie" va voir le jour. Ce cours fera une (petite) place à l'informatique et à la robotique.

· Vu ces projets de mise au chômage de nombreux enseignants et de restructuration des grilles horaires, on assiste depuis plusieurs mois à des grèves et à des manifestations des enseignants. Il n'est pas exagéré de dire que la situation au sein des écoles est proche de l'anarchie et que les professeurs sont pour le moins terriblement démotivés.

· Les budgets de la formation continue des enseignants n'ont à ce jour pas été modifiés et des initiatives dans ce domaine restent donc possibles.

Nos initiatives en matière de formation continue

· Après une année (93/94) où des groupes d'enseignants, encadrés par l'équipe du CeFIS, ont produit une série de fiches pédagogiques (voir la brochure présentant nos publications), une année (94/95) où ces fiches ont été largement expérimentées, évaluées au sein des classes et amendées, nous avons organisé cette année une formation s'adressant aux enseignants qui jouent le rôle de personnes ressources au sein de leur école, en ce qui concerne les NTIC.

· Cette formation destinée à la fois aux personnes ressources et aux enseignants d'informatique

(ce sont souvent les mêmes personnes) aborde à la fois des problèmes de didactique et d'enseignement. fournit aux participants des informations nouvelles (sur l'architecture sur de nouveaux logiciels,...) et évoque également de manière très concrète les problèmes liés à la gestion des ressources informatiques au sein de l'école (virus, réseaux, organisation d'un pool de PC, backup,...). Cette formation se poursuivra en 96/97 pour conduire à un total de 50 demi-journées de cours et T.P.

· Outre cette formation, nous avons également organisé des cours d'initiation et de perfectionnement sur le traitement de texte (Word 6 pour Windows), un cours d'initiation aux réseaux, un cours consacré au tableur Excel et un cours sur Windows 3.1.

L'INFORMATIQUE DANS LES LYCEES EN FRANCE

Jean-Pierre PEYRIN

1° Le travail du GTD Informatique de 1990 à 1993 (M. Lucas, J-M. Bérard, J-C. Boussard, D. Monasse, C. Patoux, J-P. Peyrin, R. Raynaud).

Les conclusions détaillées du GTD ont été publiées dans un rapport en juin 1993 sur l'informatique à l'école, au collège et au Lycée. Globalement, ces conclusions peuvent être résumées ainsi:

- L'utilisation raisonnée des moyens informatiques nécessite la compréhension de concepts fondamentaux de la discipline.
- L'introduction d'une nouvelle discipline, pour elle-même, n'est pas à l'ordre du jour.
- Il faut un tronc commun obligatoire regroupant la présentation des notions et savoir-faire utiles aux autres disciplines et nécessaires pour que tout citoyen puisse mesurer pleinement les apports et les enjeux de l'informatisation.

2° Une expérimentation d'un enseignement d'informatique dans les Lycées.

La DLC a décidé, en février 1994, d'expérimenter un enseignement obligatoire pour toutes les classes de secondes, suivi d'une option en première et en terminale, diversifiée selon les sections. Le comité de pilotage de l'expérimentation a proposé pour la classe de seconde un enseignement de 25h (correspondant à 10 séquences pédagogiques de 1h en classe entière et de 1h30 en demi-classes). 21 Lycées ont participé à l'expérimentation de l'enseignement en seconde et 13 Lycées participent en ce moment à l'expérimentation de deux options en première.

3° La décision n° 58 du nouveau contrat pour l'école de F. Bayrou, Ministre de l'Education Nationale.

Un enseignement optionnel d'Informatique dans les Lycées avait été institué en 1982.11 a été supprimé par le Ministère de l'Education Nationale en 1991. La décision n° 58, prise pour la rentrée 1995, recréait un enseignement optionnel d'informatique. Cette décision a été à la fois importante et catastrophique: importante, car elle officialisait, de nouveau, une présence de l'informatique dans la formation générale; catastrophique, car l'enseignement n'était qu'optionnel (donc non indispensable) et était en contradiction avec le travail accompli par le comité de pilotage de l'expérimentation. Nous avons réussi cependant à faire admettre que le contenu de l'option créée devait être celui que nous avons défini dans le cadre de l'expérimentation.

4° La rentrée scolaire 1995/96, et la suivante.

Un programme pour la seconde a été publié ainsi que des documents d'accompagnement (exemples de séquences pédagogiques, exemples de liens avec les disciplines, exemples d'évaluations). Le programme est principalement rédigé en termes de compétences attendues et évaluables; il fait une part importante au thème "informatique et monde contemporain".

En septembre 1995, ce nouvel enseignement a été mis en place: l'option devait être ouverte par tout établissement qui en avait les moyens humains et matériels et pour des classes entières (non construites par rapport à l'informatique). En septembre 1996, deux options seront ouvertes pour les classes de première: une option pour la série scientifique et une option pour les séries "lettres" et "économie-gestion". Les programmes seront publiés prochainement.

5° Les questions.

- Il faut organiser un suivi de l'enseignement effectué: le programme doit être respecté (horaire, notions, séquences pédagogiques, liens avec les disciplines).
- Il faut mettre en place une formation des formateurs et pour cela définir le contenu et la durée de cette formation. Un premier Plan National de Formation aura lieu les 6 et 7 mai pour une cinquantaine de formateurs MAFPEN.
- Il faut tendre le plus rapidement possible vers un enseignement obligatoire de l'informatique en seconde. Ceci passe nécessairement par la création d'un corps particulier pour que le Ministère puisse chiffrer le volume horaire dû par les enseignants. Nous ne souhaitons pas qu'un CAPES d'informatique soit créé, cela risquerait d'enfermer la discipline. Nous souhaitons que soit créé un CAPES de "X + informatique", où X serait n'importe quelle discipline (avec éventuellement des quotas pour chaque type de disciplines). Cela officialiserait une double compétence, seule garante des aspects transversal et ouvert de la discipline informatique.

La Situation en Suisse

Alain BRON

La Suisse avait innové en introduisant un cours d'informatique obligatoire au niveau des lycées en 1987. Les textes officiels comportaient des considérations générales ainsi que l'obligation d'offrir à tous les élèves un cours d'introduction à l'informatique. (Document 1).

A l'issue de ces décisions la CDIP (conférence des chefs de département de l'instruction publique: les 25 ministres de l'éducation en Suisse) publiait un important document. Les lignes directrices comportaient quatre chapitres principaux: données et applications, algorithmes, architectures et fonctionnement, histoire et implications. (Document 2).

La situation a changé et la mention d'un cours obligatoire d'informatique a disparu des nouveaux textes officiels. Cela pose un certain nombre de problèmes, chaque canton doit maintenant établir de nouveaux programmes d'enseignement et peut y intégrer ou non l'informatique.

Certains cantons ont conservé le statut-quo avec une ou deux périodes de cours d'informatique de base pour tous les élèves. D'autres n'ont pas encore pris de décisions.

Un lecteur d'interface nous a donné son avis plutôt optimiste. (Document 3).

Lors de réflexions sur le sujet (Forum 1995 de la CDIP et travaux préparatoires dans le canton de Vaud) quelques idées sont apparues. Elles ne cernent pas l'entier du problème, mais montre une tendance qui va plutôt vers un cours de service en partant de l'idée que des notions d'informatique sont indispensables à tous, soit pour leur travail immédiat au lycée, soit ultérieurement lors de leurs études futures et même après. (Document 4 et 5).

Un chef d'entreprise donne également son avis. (Document 6).

Ces quelques textes doivent bien sûr être complétés par le Curriculum de l'UNESCO qui est actuellement la référence en la matière.

DOCUMENT 1**ORDONNANCE SUR LA RECONNAISSANCE DE CERTIFICATS DE
MATURITE**

(Octobre 1987, Le Conseil fédéral)

Article 7

1 Le but des écoles préparant à la maturité, dans tous les types, est de donner aux élèves la maturité nécessaire aux études supérieures, c'est-à-dire de solides connaissances fondamentales et un jugement indépendant, mais non des connaissances spéciales trop poussées. Les écoles s'efforceront d'atteindre ce but en développant à la fois l'intelligence, la volonté, la sensibilité et les aptitudes physiques.

2 Les élèves du niveau scolaire final doivent être capables non seulement de comprendre, d'assimiler et d'exposer les matières qu'on leur enseigne, mais encore de saisir correctement des problèmes d'une difficulté adéquate et d'en présenter clairement la solution.

3 Les élèves possèdent un savoir sûr, un jugement indépendant et clair dans la mesure où ils sont capables de s'exprimer avec justesse et précision dans leur langue maternelle. Il est donc nécessaire d'accorder toute l'attention voulue à celle-ci non seulement dans les cours qui lui sont consacrés, mais aussi dans les autres disciplines.

4 L'école doit former des personnalités cultivées, maîtrisant les méthodes du travail scientifique et celles du traitement de l'information, aptes à travailler en commun, et en faire des membres de la société conscients de leur responsabilité d'hommes et de citoyens. Elle fera régner en son sein un esprit favorisant l'attachement aux valeurs culturelles et linguistiques de la Suisse, tout en restant ouvert sur le monde.

Article 9

6 Les élèves de tous les types doivent bénéficier d'une introduction à l'informatique. Toutes les écoles doivent en outre offrir un cours facultatif sur cette matière.

DOCUMENT 2

L'INFORMATIQUE DANS LES ECOLES DE MATURITE (DOSSIER 6) (1987, CDIP)

IDEES DIRECTRICES DU PLAN D'ETUDES

Conformément au mandat de la CDIP, le groupe de travail a établi un plan d'études basé sur une dotation de deux heures hebdomadaires pendant un an, dotation valable pour les gymnases de tous les types. La matière prévue ne pourrait pas être traitée en une heure hebdomadaire pendant un an sans restrictions sensibles. L'expérience a montré qu'une dotation encore inférieure était tout simplement insuffisante.

Nous estimons que les objectifs définis dans ce rapport devraient être atteints au plus tard à la fin de la dixième année scolaire.

L'enseignement de l'informatique a pour but de transmettre une culture générale dans ce domaine. L'élève doit pouvoir acquérir des connaissances et se faire une idée personnelle au sujet de l'emploi, du fonctionnement mais aussi des limites de l'ordinateur et de l'informatique. Un aspect important de cette formation est qu'elle tend à une formulation claire des notions dans la langue maternelle.

Le cours d'initiation à l'informatique doit donner à l'élève les connaissances et les techniques nécessaires, d'une part, à la mise en oeuvre de méthodes d'apprentissage actuelles, d'autre part, aux applications didactiques de l'ordinateur dans d'autres branches, en tirant avantage du caractère interdisciplinaire de l'informatique. L'emploi des moyens informatiques dans l'enseignement d'autres branches ne sera pas examiné ici, en dépit de son importance.

Les sujets traités dans le cours d'initiation pourront être approfondis et élaborés dans les cours facultatifs prévus par l'ORM. Le maître dispose ici d'une marge de liberté lui permettant de mettre l'accent sur des domaines particuliers. Le cours d'initiation constitue également l'une des bases de la nouvelle discipline "Mathématiques appliquées" dans le type C.

Nous proposons un cours d'initiation comprenant quatre chapitres:

1. DONNEES ET APPLICATIONS
2. ALGORITHMES
3. ARCHITECTURE ET FONCTIONNEMENT DES ORDINATEURS
4. HISTOIRE ET IMPLICATIONS SOCIALES DE L'INFORMATIQUE

Chacun des domaines peut être pondéré différemment, mais il doit être traité dans le cours d'introduction, pour conserver à celui-ci sa globalité et son pouvoir formateur et pour mettre en évidence le caractère interdisciplinaire de l'informatique. Cette exigence est parfaitement compatible avec les méthodes

pratiquées dans l'enseignement secondaire supérieur, où le maître choisit habilement ses sujets dans un ensemble plus vaste et les approfondit comme il le juge bon.

Il appartiendra aussi au maître d'adapter la pondération des quatre domaines et de choisir les exercices en fonction des exigences propres de chaque type de maturité.

Le tableau suivant présente les articulations du cours et les objectifs proposés.

OB JECTIFS

1. DONNEES ET APPLICA TIONS

L'élève doit être à même:

- a) d'expliquer la structure d'une collection systématique de données. Il devra pouvoir dire quels sont les objets mémorisés, quelles caractéristiques les représentent et quelle structure permet de les retrouver;
- b) de décrire les classes importantes de programmes d'applications (gestion de banques de données, tableurs, graphisme, traitement de textes, systèmes experts, réseaux d'ordinateurs, téléinformatique, applications en temps réel);
- c) de traiter ou d'analyser des problèmes concrets simples en utilisant l'ordinateur comme outil (à l'aide de programmes d'applications); d'expliquer la différence avec la confection d'un programme et avec le travail sans ordinateur.

2. ALGORITHMES

L'élève doit être à même:

- a) d'élaborer et de présenter systématiquement des solutions algorithmique de problèmes choisis;
- b) de formuler des algorithmes et des structures de données dans un langage de programmation évolué et de faire exécuter les programmes correspondants sur ordinateur;
- c) d'expliquer des programmes simples qu'on lui propose.

3. ARCHITECTURE ET FONCTIONNEMENT

L'élève doit être à même:

- a) de décrire le modèle d'ordinateur classique, composé d'un organe de commande, d'une unité arithmétique et d'une mémoire de travail; d'en expliquer les premiers principes de fonctionnement;
- b) de décrire les fonctions des composants et des accessoires importants d'un équipement informatique;
- c) d'expliquer la nécessité de la traduction finale d'un programme en langage machine;
- d) d'énoncer quelques propriétés importantes du système d'exploitation.

4. HISTOIRE ET IMPLICATIONS

L'élève doit être à même:

- a) de reconnaître les effets de l'ordinateur dans la vie quotidienne;
- b) de prendre conscience de certaines influences de l'informatique sur la vie familiale, sur l'école et sur les loisirs;
- c) de décrire, à partir d'exemples, les bienfaits et les dangers de l'informatique;
- d) de rapporter les épisodes marquants de l'histoire de l'informatique.

Le présent rapport pourrait porter en épigraphe: "avancer d'un pas". Les problèmes de méthode et d'enseignement de l'informatique étant, même au-delà de nos frontières, loin d'être résolus, nous n'avons pas le droit de nous borner à spécifier les objectifs à atteindre. Le développement tumultueux de l'informatique, le manque de consensus quant à la matière à enseigner et aux méthodes didactiques, les besoins en formation des enseignants, tout cela nous force à préciser notre vision de la réalisation de l'enseignement de l'informatique dans nos gymnases. Nous développerons nos idées par le biais de remarques sur les méthodes et sur la matière, ainsi que par des suggestions d'exercices.

Nous recommandons d'accorder la plus grande attention au travail autonome des élèves, aussi bien habituellement que face à l'ordinateur.

C'est intentionnellement que nous avons renoncé à faire des propositions concrètes sur la place idéale de l'informatique dans le catalogue général des branches. Il est évidemment possible de le faire soit dans le cadre d'une discipline, soit en sortant du schéma des branches classiques. Les objectifs et les contenus assignés au cours d'initiation à l'informatique montrent bien qu'il ne peut être considéré comme relatif au seul domaine des mathématiques. En conséquence, il n'est aucunement impératif d'incorporer ce cours aux mathématiques, pas plus qu'il n'est nécessaire d'en soustraire les heures exclusivement à leur dotation. Compte tenu de l'importance du caractère interdisciplinaire de l'informatique, il est souhaitable de répartir la charge horaire sur plusieurs disciplines, y compris les branches littéraires ou économiques.

DOCUMENT 3

L'enseignement de l'informatique et l'utilisation de l'ordinateur dans /es gymnases avec la RRM 95

(Tiré d'Interface 4/95) Pierre Favre, Bôle

Les intéressés ont constaté avec regret l'absence de référence explicite à la discipline ou à l'outil informatique dans le cadre de la nouvelle réglementation concernant la maturité académique. Toutefois, à bien lire les considérations générales qui définissent les objectifs de la dite maturité, on perçoit des signes qui nous montrent que cette discipline n'est pas exclue. Ainsi on dit (art. 5.1) que la maturité « les prépare à assumer des responsabilités dans la société actuelle. » Cette dernière étant largement informatisée, pourrait-on ignorer l'objet informatique dans l'organisation des études. De même (art. 5.2), on annonce que "ils se familiariseront avec la méthodologie scientifique." Cette dernière n'inclut-elle pas automatiquement des éléments liés au traitement de l'information ? Quand, enfin, on souhaite (art. 5.4) que "les élèves soient aptes à se situer dans le monde naturel, technique, social et culturel où ils vivent, ...", on voit mal comment ne pas prendre en compte la dimension informatique du monde en question.

La difficulté majeure réside dans la concrétisation de ces objectifs sans alourdir le pensum des élèves, que l'on a toujours espoir d'alléger globalement. Il est clair que chaque discipline a maintenant le devoir d'analyser dans son propre cadre la place qu'a prise l'informatique (dans toutes ses dimensions) au sein de notre société. Cela va de la philosophie à l'histoire, en passant par la géographie et l'économie. Un autre aspect est constitué par l'utilisation didactique de l'ordinateur, en allant d'éventuels logiciels EAO à des utilisations plus simples et plus directes à travers ou non ce qu'il est convenu d'appeler le multimédia. Concrétiser ces objectifs, c'est les voir apparaître explicitement dans les plans d'étude et se donner les moyens d'y accéder aussi bien en formation qu'en matériel.

Pratiquement que peut-on proposer en complément pour appuyer cette démarche et l'ancrer dans la nouvelle structure du gymnase ?

- . Un premier point concerne l'attente vis-à-vis du secondaire I. Si une introduction à l'informatique y est pratiquée en dehors d'un usage didactique de l'ordinateur, elle a certainement pour but de préparer les élèves à pénétrer dans une société informatisée à la fin de la scolarité obligatoire. Les acquis doivent alors être maintenus jusqu'à la fin de la 9ème année de scolarité et être énoncés clairement à l'intention du secondaire II.
- . Au gymnase (il ne s'agit donc plus de l'ensemble du secondaire II), les dispositions devraient porter sur l'offre de cours, le matériel et la formation des enseignants.
- . Les cours facultatifs suivants pourraient être mis à disposition des gymnasiens:
 - a. utilisation de logiciels courants (traitement de texte, tableur): mise à jour et approfondissement des acquisitions antérieures.

- b. introduction à la programmation: en vue d'une utilisation dans le domaine scientifique ou économique. Ce cours pourrait être un préalable obligatoire pour l'inscription à certaines options (applications des mathématiques, économie,...).
- c. élaboration de projets: ce cours déboucherait, pour certains, sur la préparation d'un travail de maturité.
- d. préparation à l'accès aux multimédias: quelques heures sur les manipulations techniques et les difficultés qui peuvent surgir (pourrait être aussi une partie de a.)

. Les cantons qui le souhaiteraient auraient la possibilité de présenter l'informatique comme discipline cantonale.

. Au niveau du matériel, il faudrait que l'ordinateur soit accessible immédiatement dans les locaux où l'on en a besoin: dans certaines salles d'enseignement spécialisé (physique, mathématique, biologie,...), souvent en relation avec des logiciels spécialisés, sur des chariots supportant un moniteur de grande taille et des haut-parleurs en relation avec un engin incluant un équipement multimédia, dans les locaux toujours accessibles aux élèves (en dehors de salles d'informatique), comme bibliothèque ou salle d'étude. Cela implique une réflexion à propos du renouvellement du matériel, un éventuel redéploiement de celui à disposition, ainsi que la préparation de connexions avec des réseaux de communication au fur et à mesure qu'ils deviendront performants .

• Il va sans dire que la formation des enseignants va jouer un rôle primordial pour faire face à cette nouvelle situation. Si l'on dispose pratiquement partout d'un noyau de maîtres-informaticiens capables de prendre en charge les enseignements spécialisés, il faut mettre l'accent sur la préparation des autres enseignants déjà en place. En dehors de cours d'initiation et de réflexion, il faut surtout leur assurer un appui permanent dans les établissements, qui évitera d'allonger démesurément la formation stricte et leur assurera une sécurité indispensable. Il est clair, par ailleurs, que la formation initiale des enseignants de toute catégorie inclut une telle préparation.

En conclusion, il nous semble que la place de l'informatique et de l'utilisation didactique de l'ordinateur est à prendre dans le cadre des applications de la RRM 95. Il est toutefois nécessaire de trouver rapidement des solutions réalistes et de les inclure dans l'ensemble des réflexions menées dans chaque canton pour construire la nouvelle maturité académique. Il faut avoir foi en son idéal et se garder de réactions épidermiques et maladroités, qui ne conduiraient qu'à éliminer une réflexion et une préparation à un phénomène omniprésent dans notre société actuelle et à venir.

DOCUMENT 4**Quelques réflexions en vrac.**

1° Les logiciels qu'utiliseront nos élèves n'existent pas encore. Il est donc important de leur apprendre à découvrir par eux-mêmes un nouveau logiciel en insistant sur les notions fondamentales qui se rencontrent toujours.

2° Ce n'est pas l'usage du tableur X ou Y qui est important, mais les concepts sous-jacents (notion de case, de formule, de test, ...)

3° Ne pas oublier que l'ordinateur est un exécutant, il faut donc lui donner des ordres (par exemple si on veut centrer un titre, il ne faut pas taper des espaces blancs, mais demander à l'ordinateur d'effectuer le travail).

Pour le traitement de texte par exemple:

- Règles de ponctuation,

- Inutile d'apprendre que c'est CTRL Machin Pomme au lieu de Alt Truc qu'il faut utiliser pour sauvegarder son texte. L'important, ici, c'est de comprendre l'utilité d'une sauvegarde régulière qu'elle soit automatique ou manuelle.

- Passer plusieurs mois à s'essayer à la dactylographie pour améliorer la vitesse de frappe est moins important* que d'apprendre à chercher, trouver, classer, trier, organiser les informations (scannées, sur CD, sur un serveur, ...) et apprendre à créer son propre document. Mais un cours facultatif de dactylographie pourrait être proposé s'il n'a pas été offert auparavant.

* d'autant plus que dans quelques années les logiciels de reconnaissance vocale auront fait d'énormes progrès. Actuellement déjà ils augmentent la productivité, mais nécessitent encore un apprentissage de plusieurs heures.

4° Si le travail demandé à l'élève au cours d'informatique est en relation directe avec une activité dans une autre discipline, son efficacité sera accrue.
Notion d'interdisciplinarité.

5° Un enseignement de l'informatique est formateur (notion de rigueur, de précision, ..) mais l'aspect service est primordial (pour un enseignement de base pour tous les élèves).

6° Rendre l'élève autonome (indépendant) plus que performant.

7° Notion de partenariat. C'est en accord avec les partenaires qu'un programme d'informatique doit être élaboré:

Partenaires horizontaux: disciplines parallèles.

Partenaires verticaux: les enseignants (écoles) situés en amont et en aval.

Partenaires extérieurs: le monde du travail, les parents et pourquoi pas les élèves ? (Lors d'une rencontre " maîtres-élèves n ceux-ci demandaient que l'école s'ouvre plus sur la vie réelle).

8° Le cours d'introduction à l'informatique, avec sa faible dotation horaire (15- 16 séances), ne permet de faire que l'essentiel.

9° On doit préparer un programme pour des élèves qui entreront dans la vie active dans 7 ou 8 ans. Si le programme doit durer une mutation, soit environ 10 ans, on peut donc se fixer 2010. S'il est facile de dire ce que sera la langue allemande dans 15 ans, il n'en est pas de même avec l'informatique. On est dans la situation suivante: c'est comme si on se situait en 1980 et que l'on devait préparer un programme pour 1995. En 1980 on ne parlait pas de bureautique (selon le Robert: ensemble des techniques (informatique, télématique ...) visant à automatiser les travaux de bureau) et qui aurait imaginé que le fax deviendrait incontournable ? Mais en 1980 Internet existait. Et maintenant qui peut dire ce que sera le concept de bureautique en 2010 ? Par contre les liaisons entre ordinateur seront encore d'actualité. Les ordinateurs auront encore une architecture Von Neumann, mais je pense qu'on leur parlera (certain songe à autre chose qu'un ordinateur, mais les coûts de conception étant tellement élevés que je doute que l'on ait remplacé une architecture qui donne satisfaction depuis 50 ans). Il est probable que la notion de texte (avec un traitement de texte), de tableau (avec un tableur) aura disparu au profit d'une notion générale de document.

10° Deux objectifs principaux:

- déterminer ce qui est constant en informatique (par exemple la notion de codage de l'information, la notion de document, ...),
- déterminer ce qu'il faut faire pour rendre les élèves indépendants, pour qu'ils puissent pendant leurs études être autonome et qu'ils puissent découvrir sans trop de difficultés un logiciel nouveau.

DOCUMENT 5**Nouveau programme d'enseignement de l'informatique au secondaire II**

(proposition d'octobre 1995 dans le canton de Vaud)

Introduction

L'informatique est en évolution rapide, il suffit pour s'en convaincre d'ouvrir un journal. Partout on ne parle que de réseaux mondiaux d'information, de CD-ROM et de banques de données.

La nouvelle maturité insiste beaucoup sur l'interdisciplinarité et prévoit un travail

personnel où l'exploitation des sources de données électroniques est extrêmement précieuse.

Pour compléter les objectifs du programme précédent (comprendre et maîtriser un environnement informatique, analyser et mettre en oeuvre les moyens informatiques adaptés à ses problèmes), il nous semble indispensable, à l'heure où l'on parle d'autoroutes de l'information, de permettre aux élèves de comprendre et d'utiliser ce formidable moyen de documentation et de communication. (Cf. *Informatics for Secondary Education*, UNESCO 1994)

C'est pourquoi nous vous proposons un nouveau programme.

1. Objectifs**Objectifs généraux**

- Donner à l'élève une vision critique de l'informatique, en particulier envers les différentes sources d'information.
- Permettre à l'élève d'utiliser de manière autonome les moyens informatiques et les sources extérieures de renseignements pour pouvoir les intégrer dans différentes activités scolaires.

Justifications et fondements

- La gestion de l'information est un phénomène de société.
- L'attitude critique envers une source d'information et son exploitation sont deux aspects fondamentaux de la vie moderne et il est indispensable de les développer.

Objectifs didactiques

L'élève est capable de:

- comprendre et maîtriser un environnement informatique
- comprendre et utiliser différentes formes de banques de données
- analyser et mettre en oeuvre les moyens informatiques adaptés à ses problèmes
- réunir et organiser les informations adaptées au but de sa recherche
- créer des documents intégrant des informations extérieures.

BON DE COMMANDE

Je souhaite recevoir _____ exemplaire(s) des Actes de la

Cinquième Rencontre Francophone sur la Didactique de l'Informatique

Nom:..... Prénom.....

Adresse:.....

.....

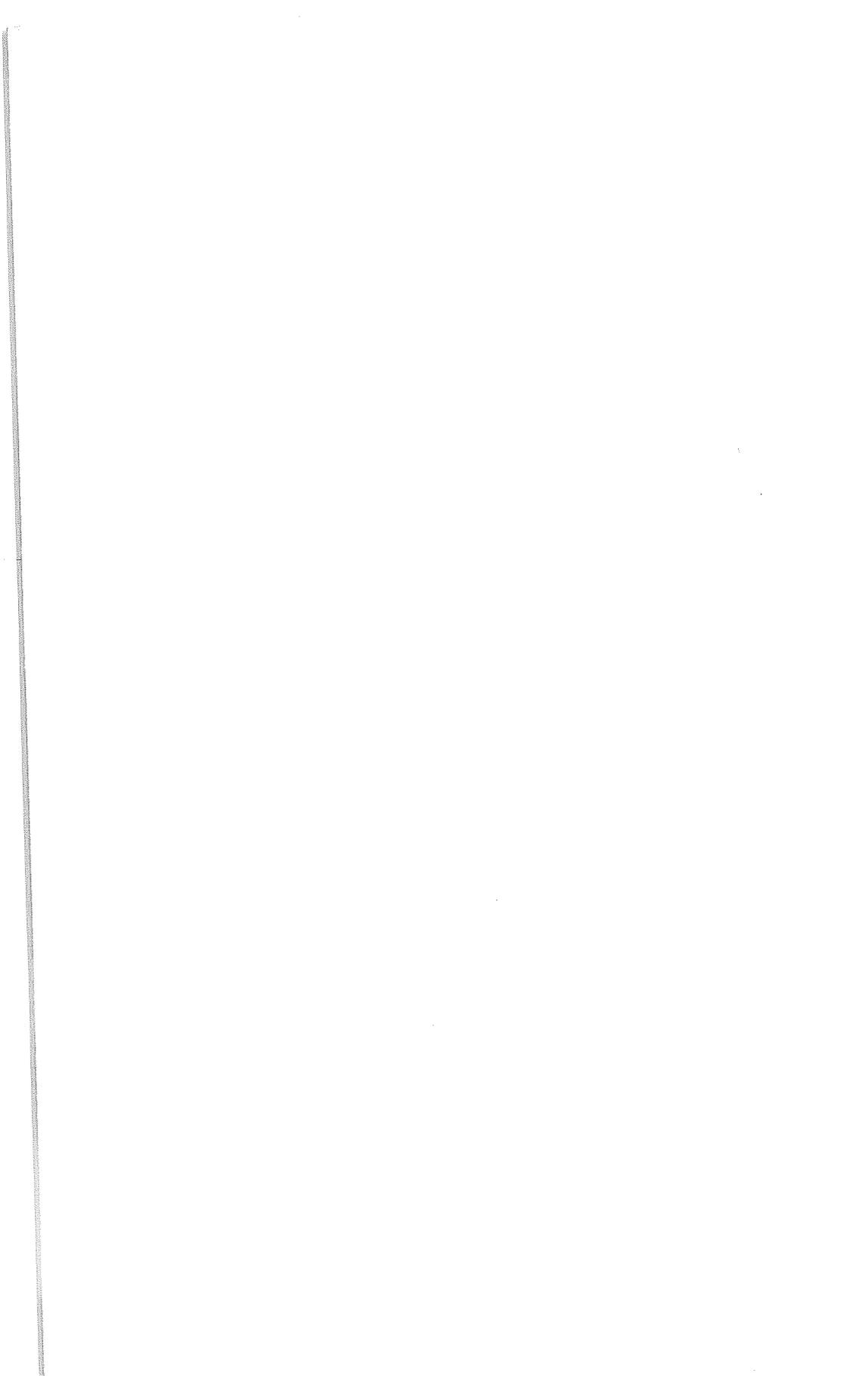
..... Ville.....

Ci-joint un chèque (ou bon de commande)

*Le prix de l'exemplaire est de 120 Francs Français en TTC ou
équivalent en d'autres monnaies*

COMMANDE A ADRESSER :

- En Tunisie: Institut National de Bureautique et de Micro-Informatique -INBMI-
18 bis, rue de l'Inde - 1002 - Tunis - Tunisie
- En France: EPI - 13 rue du Jura - 75013 Paris
- En Suisse: SSIE - Case Postale 588 - CH 1401 YVERDON
- En Belgique: CeFIS Facultés N-D. de la Paix - rue de Bruxelles, 61 5000 Namur



Quel rôle peut-on conférer à la matière informatique scolaire dans les pays en voie de développement? Les expériences vécues ainsi que leurs évaluations ont souvent concerné les pays développés. Dans quelle mesure les résultats de ces évaluations restent-ils valables pour les pays en voie de développement?

La démocratisation de l'outil informatique conjuguée à une sensibilisation solide à la culture informatique est à même d'assurer un transfert technologique approprié au contexte socio-culturel et de garantir un bon usage raisonné de cet outil.

Dans les pays développés, l'informatique s'impose comme un phénomène de société, et ce dans toutes les sphères socio-économiques (administrations, banques, gares, supermarchés, usines, foyers, etc.) Ainsi, on est mieux familiarisé dans ces pays, à la pratique informatique par des usages extra-scolaires même en l'absence d'une discipline informatique scolaire.

Dans les pays en voie de développement, la démocratisation de l'outil informatique, bien qu'elle soit dans leurs préoccupations, est loin d'être réalisée. Les pouvoirs publics sont amenés à jouer un rôle précurseur et prépondérant dans la diffusion de cette culture. Ces pays auraient à envisager une introduction réfléchie et adéquate de l'informatique dans les réformes de leur système éducatif pour la faire sortir de son cadre expérimental, en lui attribuant, entre autre, un statut de matière enseignée, ce qui permettrait de lui accorder les priorités requises et les moyens adéquats et de relever le défi pour généraliser l'accès aux technologies d'information et éviter qu'elles ne deviennent des facteurs d'exclusion.

La communauté scientifique internationale a, fort heureusement, rompu son silence en promulguant, à l'intention des pays en voie de développement, des recommandations consistantes, dans les *Résolutions* des différents congrès internationaux.

Tahar HAFSAIED

Président de l'AFDI

*Imprimerie Officielle
de la République Tunisienne*

1996

I.S.B.N. 9973-906-63-2