

CHAPITRE 7

LES ASPECTS DIDACTIQUES

La nécessité d'une approche didactique n'est plus à démontrer, compte tenu des nombreuses difficultés dont nous venons de faire état...

Mais ces questions ne sont pas simples. La didactique n'est pas une réflexion unitaire, elle dépend étroitement des disciplines et des contenus. En fait, on ne peut pas parler d'une didactique, mais **des didactiques des disciplines**. Et pour nous, point de discipline (au sens strict du terme).

Nous nous voyons donc contraints de voyager à travers des éléments disparates, et d'emprunter aux didactiques disciplinaires des notions susceptibles d'être adaptées à notre domaine.

C'est ainsi qu'après un point théorique rassemblant succinctement les principaux concepts élaborés par les didacticiens des disciplines « classiques », nous présenterons les réflexions d'un enseignant sur un cadre didactique général, élaboré à partir d'un stage MAFPEN à destination des Inspecteurs Pédagogiques Régionaux. Puis l'expérimentation de l'enseignante de Toulouse à propos des relations mémoire-écran sera reprise dans ses aspects proprement didactiques (le « modèle pédagogique » de B. MARI-BARTH). Enfin, l'équipe de Rouen présente une approche des objets et des logiciels à partir des Langages Orientés Objet, exemple intéressant de l'utilisation des théories de la « science informatique » à l'usage pratique des ordinateurs.

CADRES THÉORIQUES

Nous ne ferons pas ici un exposé exhaustif sur l'ensemble des concepts des didactiques¹ ; nous nous contenterons de relever les principales questions que nous posent l'absence d'un cadre théorique abouti ; nous citerons ensuite des notions que nous pouvons emprunter et adapter à nos contenus.

1. Lire LEVY J.-François, *op. cité*, et plus particulièrement le chapitre 2 « Eléments de théorie », dans lequel nous développons cette partie.

Histoire et épistémologie de l'informatique

Les références des savoirs

Les didacticiens des disciplines à « références savantes » (mathématiques et une partie des sciences physiques) se réfèrent souvent avec profit à l'épistémologie de leur domaine pour élaborer des concepts fondamentaux à partir de leurs contenus. Nous en sommes partiellement privés, car l'histoire de l'informatique² manque de recul pour prendre en compte les avatars récents des micro-ordinateurs, et le domaine proprement dit de la « science informatique » concerne des théories qui, comme nous l'avons déjà signalé, n'ont pas souvent de relations directes avec nos pratiques.

En conséquence, nous ne pouvons pas utiliser le concept de « *transposition didactique* », élaboré par CHEVALLARD³, en mathématiques. Rappelons que ce terme désigne la transformation qui s'opère entre le savoir tel qu'il s'élabore dans les milieux scientifiques, pris comme référence, et l'état de ce savoir transmis par l'institution scolaire

Le concept de « *pratique socio-techniques de référence* » semble mieux adapté à notre domaine: J.-L. MARTINAND définit les situations de *référence* dans le domaine professionnel, axées sur des *pratiques à caractère technique*, dans des secteurs *sociaux* bien spécifiés. Ce concept permet de mieux aborder les enseignements et formations technologiques et professionnels dispensés dans les établissements et lieux spécialisés.

Si ce concept se rapproche davantage de nos activités, il soulève cependant quelques problèmes :

- les lieux de pratique sont multiples : contrairement à l'exercice d'un « métier classique » industriel ou tertiaire, socialement situé, l'utilisation du micro-ordinateur se diffuse dans toute la société ;
- les buts ne sont pas identiquement définis : il n'y a pratiquement aucun point commun entre le « bricolage domestique » sur traitement de texte (ce sont les connaissances - très empiriques - de l'utilisateur qui définissent les buts accessibles) et l'utilisation d'une station de Publication Assistée par Ordinateur (PAO) chez un éditeur.

De plus, ces situations sont actuellement en évolution rapide et en partie imprévisible. Si nous pouvons miser sur une diffusion de plus en plus importante de la micro-informatique dans l'ensemble de la société, nous ne savons pas comment le public s'adaptera concrètement, ni ce qu'il en fera - pratiquement et intellectuellement...

La transmission des savoirs

Les représentations et conceptions préexistantes, déjà évoquées dans le chapitre 2, posent d'importants problèmes didactiques ; elles sont directement liées à la notion d'« *obstacle épistémologique* », notion spécifiée par G. BACHELARD au milieu du siècle : les conceptions erronées qu'avaient les

2. Lire BRETON P. [1990] *Une histoire de l'informatique*, Paris, Points science, Seuil.

3. Lire CHEVALLARD Y. [1985 2ème édition 1991]. *La transposition didactique, du savoir savant au savoir enseigné*, Grenoble, La pensée sauvage.

étudiants de certaines lois physiques préexistant à leur enseignement s'opposaient, parfois de manière durable, à leur assimilation⁴.

Beaucoup plus récemment, J.-L. MARTINAND⁵ propose la notion « **d'objectif-obstacle** », notion didactique dans laquelle la mise en évidence de l'obstacle et la recherche des moyens de le surmonter doit être considérée comme une démarche formatrice particulièrement efficace.

Ce concept amène à redéfinir la notion de tâche: au plan didactique, une *tâche* se caractérise par les *obstacles* qu'elle implique d'affronter et qui constituent les *objectifs* pour lesquels cette tâche est proposée aux apprenants⁶.

UNE PROPOSITION DE CADRE DE RÉFLEXION DIDACTIQUE À PARTIR D'UN STAGE DE FORMATION

Un enseignant de philosophie de Nancy⁷ a mis en place une formation MAFPEN pour un public d'Inspecteurs Pédagogiques Régionaux (IPR), dont l'objectif était de faire améliorer rapidement la pratique du traitement de texte. A cette occasion, et à partir des hypothèses qu'il avait émises lors du travail sur CONNEXIONS (voir chapitre 6), il propose un cadre de réflexion didactique dans lequel on peut situer contenus de formation, formateur et stagiaires.

⁸ Objectifs du stage

Des IPR (Lettres, Histoire et Géographie, Langues, Mathématiques) et une personne responsable du secrétariat de l'Inspection ont demandé au formateur de définir une approche où convergeraient d'une part **des réponses pratiques** aux besoins les plus immédiats en matière de traitement de texte et d'autre part **des éléments de nature conceptuelle** sur le fonctionnement de l'ordinateur (en tant qu'outil de traitement de texte). Le stage doit durer douze heures.

Ce public est réputé débutant en matière de traitement de texte (à une exception près, novice), sauf pour le clavier de la machine à écrire, ce qui ne va pas sans poser certains problèmes de passage d'une pratique technique à l'autre⁹.

4. Lire : G. BACHELARD [1983]. *La formation de l'esprit scientifique*, Vrin, Paris.

5 Lire : MARTINAND J.L. [1986]. *Connaître et transformer la matière, des objectifs pour l'initiation aux sciences et techniques*. Berne (Suisse), Peter Lang.

6. M. DEVELAY.

7. C. EURIAT.

8. Contribution C. EURIAT.

9. Lire LEVY J.F. op. cité.

Les besoins peuvent être formulés de la façon suivante¹⁰ :

- Savoir modifier un document existant, dans son contenu comme dans sa présentation ;
- Savoir saisir un nouveau document (document administratif ou pédagogique bref) sur le papier à en-tête de l'Inspection ;
- Savoir sauvegarder un document.

Un certain nombre de fonctionnalités comme la correction d'orthographe, la recherche de synonymes ou le publipostage demandées à titre informatif ont été explorées rapidement (soit sous forme de « récréations », soit en fin de formation).

Construction de la formation

Selon nos hypothèses et nos observations précédentes (voir chapitre 6, stage CONNEXIONS), le public du stage serait enclin - compte tenu de sa demande - à utiliser une démarche de rationalisation plutôt que des tâtonnements empiriques. C'est pourquoi nous pensons qu'il serait intéressant d'exploiter de telles dispositions, d'une part sur le plan de la rationalisation, en favorisant la construction de cette *rationalité fonctionnelle* posée comme hypothèse de fonctionnement dans le chapitre précédent, d'autre part sur le plan de la *motivation*, parfaitement identifiable et homogène ici (ce n'est pas toujours le cas dans d'autres formations dans lesquelles des motivations existent, mais sont hétérogènes et/ou difficiles à caractériser).

Motivation

En ce qui concerne la *motivation*, on peut dire qu'elle contribue à *donner du sens* à l'apprentissage, dans la mesure où elle est une motivation interne. Nous entendons par là une motivation qui relie directement la compétence à acquérir à un besoin de l'apprenant, par opposition à des motivations externes comme les notes, les examens, les décorations et autres carottes et bâtons. Dans la situation dont il est question ici, ce sens est en principe donné au départ, la demande étant clairement exprimée en fonction d'un besoin dont on peut assez facilement vérifier la réalité. Toutefois, on oublie trop souvent d'entretenir ce lien et les apprenants perdent parfois le sens de ce qu'ils sont en train de faire, en particulier quand apparaissent des difficultés qui mobilisent la quasi totalité des ressources intellectuelles disponibles sur le moment.

¹⁰ Les formés disposent d'un secrétariat. Toutefois, en raison notamment des horaires tardifs qu'ils pratiquent souvent, ils souhaitent pouvoir se servir eux-mêmes d'un traitement de texte.

Il me semble qu'il est de la responsabilité du formateur d'aider à **établir ce lien du sens** entre les motivations et les contenus de l'apprentissage en présentant **explicitement** en quoi il consiste dès le début de son intervention. Un moyen peut être de faire reformuler par les stagiaires leurs besoins avant de les laisser toucher aux ordinateurs. C'est ce que j'ai fait avec les personnes présentes à ce stage. Nous nous sommes aperçus que ce n'était pas inutile, même si la commande avait été passée d'une façon tout à fait précise par un responsable¹¹.

Il est également de la responsabilité du formateur de **maintenir la tension de ce lien** tout au long de la formation. Il faut, au cours de la formation, chaque fois qu'un nouveau problème se présente, donner à chacun l'occasion de se repositionner par rapport à ses utilisations futures et donc par rapport à des situations dans lesquelles il peut aisément se projeter. Cependant, il peut apparaître une difficulté à faire pressentir en quoi l'utilisation du nouvel outil pourra transformer ces situations par rapport aux pratiques de référence antérieures.

Le formateur a besoin d'identifier le mieux possible une sorte de panoplie de situations ayant un sens pour les stagiaires et imaginer en quoi l'introduction du nouvel outil pourra les faire évoluer (ce qui ne va nécessairement de soi)¹². L'évocation de ces situations constitue une des dimensions de l'espace¹³ de formation dans lequel on place les stagiaires. La "panoplie" constitue une ressource pour le formateur. Ce qui suppose de sa part une connaissance convenable des activités professionnelles des stagiaires.

Disons que pour le formateur cette dimension prend la forme d'une préoccupation, celle du **sens de la compétence à acquérir**.

11. En fait, il y a déjà là un petit travail sur les représentations qui mériterait sans doute d'être approfondi.

12. Car il y a à lutter contre l'idée reçue selon laquelle un outil est axiologiquement tout à fait neutre et que son usage seul détermine sa valeur. C'est oublier simplement qu'un outil est lui-même le résultat de l'usage d'autres outils.

13. Nous ne parlons pas de dispositif car ce mot renvoie finalement (même si l'on s'en défend) à une approche comportementaliste de la formation. Nous n'en faisons pas une question de religion, le paradigme behavioriste peut sans doute être pertinent pour certains apprentissages, mais il convient mal, semble-t-il, à l'acquisition de compétences complexes et susceptibles d'une évolution autonome. Par ailleurs, le dispositif y est une cause déterminante de l'apprentissage qui s'impose en quelque sorte à l'apprenant. Comme le dispositif est conçu par un formateur, on aperçoit bien que ce dernier ne s'efface derrière lui que pour mieux exercer un pouvoir sur l'apprenant, passif et malléable par hypothèse. Si le formateur est de bonne foi, il n'en est pas moins le transmetteur autoritaire d'un mode de penser et d'agir. Ce petit problème éthique vient s'ajouter au précédent. Plutôt que dispositif, nous préférons employer le mot espace. Un espace de formation est un ensemble de conditions et de ressources mises en place en vue de favoriser l'acte d'apprendre qui reste à l'initiative de l'apprenant. Ces conditions ne sont jamais posées comme nécessaires et encore moins comme suffisantes.

Rationalisation

En ce qui concerne la *rationalisation*, il est possible de jouer de la conscience spontanée de l'existence d'une logique informatique du produit que l'on remarque chez les stagiaires (Cf. hypothèses à partir du stage CONNEXIONS). Cette conscience est alors plutôt confuse.

En fait, il faut savoir si la compétence à acquérir relève réellement des seuls savoir-faire de manipulation du logiciel. Il y a, semble-t-il, au moins deux raisons d'en décider autrement. L'une d'ordre didactique : Un apprentissage des seuls comportements dans un schéma Stimulus/Réponse ne prépare pas à l'autonomie ultérieure (en particulier face aux modifications des versions successives des logiciels). L'autre d'ordre humaniste : le stagiaire appartient à l'espèce Homo Sapiens !

Quand elles sont pensées par l'utilisateur, les manipulations prennent la forme *d'actions à entreprendre dans le logiciel*, avec ce que cette expression suppose d'engagement et de conscience de la part de l'« acteur ».

Dans le cas qui nous occupe, les apprenants avaient demandé clairement des « explications théoriques ». Nous avons mis ces dispositions à profit pour fournir, sous la forme d'un exposé en début de formation, un certain nombre de notions générales d'informatique (architecture de l'ordinateur, codage et flux des données, fichiers, etc., qui correspondent à des éléments d'une liste de concepts que nous avons pointés au cours de la recherche). Ces notions constituent un ensemble de *concepts informatiques sous-jacents* au fonctionnement d'un outil.

Dans la mesure où l'on prend soin d'établir le lien de sens avec les situations d'utilisation et avec les actions à entreprendre dans le logiciel, cette première approche par les concepts généraux ainsi exposés au début de la formation ouvre la voie d'une rationalisation que l'on pourrait qualifier de fonctionnelle.

On remarque que les explications fournies ou les comportements choisis prennent en compte d'une manière souvent très intimement liée la logique de l'utilisateur qui apparaît comme dominante et relativement claire d'une part, et celle de l'informatique qui se présente comme une contrainte restrictive et plutôt mystérieuse d'autre part. Or la rationalité informatique de l'outil intègre des dimensions techniques, historiques, commerciales, voire franchement anecdotiques qui obligent parfois à des détours ou à des artifices dans les explications proposées. Il nous semble que ce qui est important, c'est que l'apprenant se construise un modèle cohérent du fonctionnement de l'outil (machine et logiciel) qui soit aussi

conforme que possible à celui qu'aurait un spécialiste du sujet, sans pour autant s'empêtrer dans les notions techniques¹⁴.

Il manque une dimension. Pour assurer la cohérence de l'apprentissage et pour tenir compte de la tendance remarquée chez les stagiaires à procéder à ces sortes de vérifications "expérimentales" des effets des actions entreprises, il convient d'inviter à regarder ce que fait effectivement l'ordinateur quand on lui envoie une commande. Cela peut paraître évident, voire trivial, mais en fait les utilisateurs ne procèdent à des vérifications qu'à la fin du travail, ou d'une partie du travail qui ne correspond pas à un découpage ayant un sens par rapport au fonctionnement de la machine. Par ailleurs, il y a des cas où rien n'est évident. Si un novice peut assez spontanément remarquer le clignotement du témoin optique du disque dur par exemple, il n'est guère probable qu'il pense tout seul à sauver un fichier au format ASCII et à le rappeler ensuite dans le traitement de texte pour saisir la différence entre le texte (caractères) et ses attributs (mises en forme diverses).

Cette dimension de la lisibilité du fonctionnement de l'ordinateur est souvent laissée de côté aujourd'hui. Soit qu'on veuille se démarquer des pratiques d'une époque « bidouillante » qu'il est de bon ton d'évoquer avec condescendance, soit qu'on tombe dans le panneau des interfaces graphiques prétendument transparentes¹⁵ avec leurs icônes et fenêtres.

Propositions d'ordre théorique

La prise en compte des hypothèses formulées précédemment et les réflexions ci-dessus nous conduisent à définir les quatre dimensions de l'espace dans lequel le formateur invitera le stagiaire à entrer et qu'il lui faudra prendre en compte au moment de la conception d'une formation. C'est de cette manière qu'a été réalisée la formation des IPR à laquelle nous nous référons ici. Nous n'avons guère d'indicateurs d'évaluation en dehors de l'expression de leur satisfaction, et aussi du fait qu'ils (pas tous et à des degrés divers) utilisent d'une façon autonome l'outil dans le cadre de leur travail.

Ces quatre dimensions sont présentées ici de trois façons : une simple liste, un schéma et un tableau.

La liste montre que les quatre dimensions correspondent à des **actions** conjointes du formateur et des stagiaires. Chacun des verbes employés mériterait peut-être un développement ultérieur.

14. Au fond, a-t-on vraiment besoin d'assumer le problème de la concordance entre la logique de fabrication de l'outil et sa logique de fonctionnement ?

15. A propos de ce mot « transparent », nous trouvons son emploi dans le petit monde de l'informatique particulièrement peu adapté : opaque conviendrait plutôt à sa place. Nous revendiquons pour les apprenants une transparence véritable, c'est à dire la propriété d'une matière à travers laquelle on voit les choses qui sont derrière.

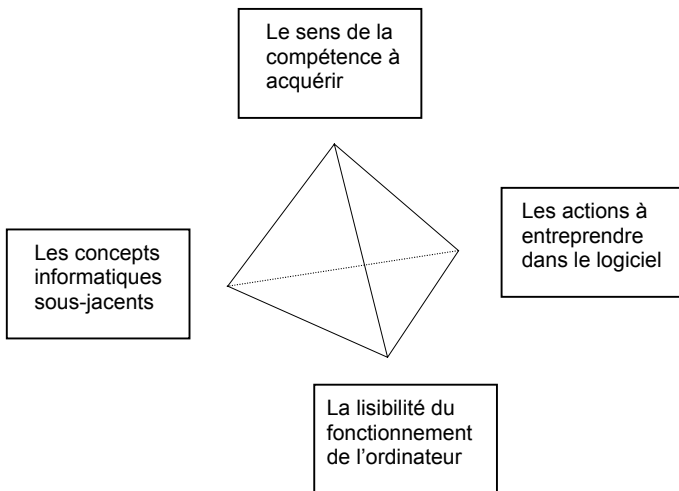
Le schéma met en évidence les liens entre les dimensions, matérialisées par les sommets d'un tétraèdre. On voit bien qu'il s'agit d'un espace d'apprentissage. On voit aussi les tensions qu'il faut établir et maintenir en portant son attention à chacune des six arêtes. Cela ne rend toutefois pas compte de la dimension temporelle qu'il ne faudrait pas oublier pour autant.

Le tableau montre par quelques exemples concrets comment on peut mettre en œuvre ce modèle.

Liste

- Construire et entretenir le sens de la compétence à acquérir par rapport aux besoins.
- Préciser les actions à entreprendre dans le logiciel pour faire appel à ses fonctionnalités.
- Mobiliser les concepts informatiques sous-jacents au moment opportun (c'est à dire à chaque fois qu'ils sont mis en œuvre dans le fonctionnement du logiciel).
- Assurer la lisibilité du fonctionnement de l'ordinateur.

Schéma montrant les quatre dimensions de l'apprentissage d'un outil informatique



Le formateur met en place les quatre sommets de cet espace d'apprentissage.

Il aide ensuite l'apprenant à créer et à maintenir la tension entre eux.

L'apprenant doit toujours se trouver "à l'intérieur" du tétraèdre.

Selon les moments, il se trouve plus ou moins proche de l'un ou l'autre des quatre sommets, mais il n'en perd jamais aucun de vue.

Quel que soit le sommet sur lequel on porte momentanément l'attention, il convient de s'assurer que l'apprenant fait le lien avec les trois autres.

C. Tableau de correspondance des quatre dimensions de l'apprentissage d'un outil informatique (exemples).

Le sens de la compétence à acquérir Besoins/ Motivations	Les actions à entreprendre dans le logiciel Menus dans WORD	Les concepts informatiques sous-jacents	La lisibilité du fonctionnement de l'ordinateur Moyens d'améliorer
Retrouver un document sur disque ou disquette.	Activer le Menu FICHIER OUVRIR. Déplacements dans les répertoires et changements de lecteurs.	Fichier. Répertoire.	Témoins de l'activité des lecteurs (bruit de moteur, LED).
Accéder à un document.	Activer le Menu FICHIER OUVRIR.	Stockage de l'information. Codage de l'information. Mémoire de travail.	Basculement sous DOS ou gestionnaire de fichier pour y lire les noms, les tailles, les dates ou les heures des fichiers.
Créer un nouveau document.	Activer le Menu FICHIER NOUVEAU DOCUMENT. MODÈLE.	Mémoire de stockage. Flux de données.	
Modifier le contenu d'un document.	Taper au clavier. COUPER, COLLER par le Presse-papiers.	Codage des caractères. Mémoire de travail. Affichage.	Saisir quelques caractères avec la combinaison de touches ALT Code ASCII. Saisir en « aveugle » (le curseur texte n'étant pas à l'écran) et faire défiler pour constater le résultat.
Modifier la présentation d'un document.	Menu FORMAT CARACTÈRE. Menu FORMAT PARAGRAPHE. Menu FORMAT BORDURE. Menu INSÈRE SAUT DE PAGE.	Codage des attributs des caractères. Codage des attributs des paragraphes.	Sauver un document enrichi (couleurs, bordures) au format texte, le fermer et le rouvrir.

L'EXPÉRIMENTATION D'UN MODÈLE PÉDAGOGIQUE

Lors de son expérimentation sur les rapports mémoire-écran (voir chapitre 5), l'enseignante de l'équipe de Toulouse¹⁶ a mis en place une séquence de formation inspirée des travaux de B. MARI-BARTH¹⁷. Nous en rapportons ici le volet didactique.

Un point de théorie

S'interrogeant sur les difficultés à enseigner l'abstraction dans la démarche générale de construction du savoir, B. MARI-BARTH élabore des éléments théoriques et une stratégie érigée en « modèle pédagogique d'apprentissage » autour des notions de **concept, de conceptualisation et d'attribut**.

Son ouvrage pose les bases de sa théorie, illustrée par des exemples pris pour beaucoup dans les sciences de la nature. Ce domaine d'exemple et la manière dont elle l'utilise nous fera émettre quelques réserves quant à l'usage que nous pouvons en faire en informatique, car l'aspect descriptif (la préoccupation taxinomique est dominante chez les naturalistes) ne permet pas, à notre avis, de bien prendre en compte les interactions apprenant-dispositif, fondamentales dans les activités avec instruments.

Le concept, notion classificatoire, est défini de manière opératoire par un nom (étiquette), des attributs et des exemples.

Les attributs sont des qualités distinctives entre idées, objets et donc concepts. Les attributs essentiels permettent le discernement premier, les autres sont moins importants (moins discriminants).

Le niveau de complexité d'un concept est fonction du nombre d'attributs qui le spécifient.

Le niveau d'abstraction est caractérisé par le champ couvert et l'éloignement aux éléments concrets. En trouver une valeur pertinente résulte d'un compromis entre ces deux pôles. Un autre élément qui fonde la validité d'un concept est l'équilibre entre les aspects objectifs (indiscutables) et subjectifs (empiriques).

Un ensemble d'attributs est caractéristique d'un concept. Les combinaisons de concepts forment des règles et des théorèmes, qui, regroupés à leur tour, constituent des théories.

La conceptualisation est la formation, l'acquisition des concepts chez l'apprenant. Il s'agit de savoir discriminer, associer, rejeter des attributs selon un cadre de référence et des structures cognitives.

Le « **modèle pédagogique** » proposé consiste à créer une situation de classe basée sur une discussion collective entre l'enseignant et tous les participants, par laquelle émergent des attributs dont on discute la pertinence pour définir le concept objet de la leçon. Après avoir grossièrement séparé les attributs pertinents (« exemples OUI ») des autres (« exemples NON »), on affine la

16. D. MARTY.

17. Lire MARI-BARTH, M. [1987]. *L'apprentissage de l'abstraction*, Paris, Nathan. MARI-BARTH, M. [1993]. *Le savoir en construction*, Paris, Retz.

classification en montrant leurs caractères plus ou moins essentiels. Une synthèse finale (institutionnalisation) permet de mémoriser à long terme un contenu stabilisé.

18 Revenons à l'expérimentation

Dans le cadre du cours de Technologie, les élèves doivent être capables à la fin du cycle d'observation d'identifier les différents constituants d'un système informatique ; mais la compétence poursuivie en priorité est la mise en oeuvre de traitements de calculs et de textes.

Les définitions des mémoires telles qu'elles sont proposées dans différents types d'ouvrages (scolaires, scientifiques ou de vulgarisation) relèvent plutôt de typologies : comparaisons entre les différentes mémoires (morte, vive et de masse, permanente ou temporaire, internes ou externes) et ne correspondent pas de façon assez précise aux caractéristiques fonctionnelles.

Nous avons donc essayé de résoudre ce problème en recherchant d'abord les attributs communs à toutes les mémoires informatiques. Il a fallu alors envisager de traiter le **concept d'information** pour différencier les deux types d'information que sont les données et les instructions.

Ce choix est renforcé par les résultats du dépouillement des questionnaires auxquels ont répondu les élèves de troisième l'année précédente et qui montraient que **la notion de programme** était en effet très lacunaire dans leurs conceptions.

Les attributs essentiels ont été mis en évidence dans le cadre de tâches du niveau sixième et cinquième utilisant les logiciels de traitement de texte Works et de dessin assisté par ordinateur Sketch.

Les attributs d'une mémoire informatique :

- élément matériel du système informatique ;
- partie assurant la fonction de mémorisation (contenir, conserver et restituer des informations).

L'information :

Une information est soit une donnée, soit une instruction (ordre)

- Les attributs d'une donnée :
 - une donnée qu'elle soit du texte, des nombres, des dessins... est entrée par l'utilisateur ;
 - une donnée peut être traitée par l'exécution des instructions d'un programme.
- Les attributs d'une instruction :
 - une instruction est un ordre faisant partie d'un ensemble d'ordres (le programme) entrés par le concepteur du programme, activés par l'utilisateur au moyen des commandes ;
 - l'utilisateur (au niveau où nous travaillons) ne peut utiliser que les commandes proposées par le logiciel ;
 - une instruction en cours d'exécution « traite » (agit sur) les données sélectionnées (par défaut ou par l'utilisateur).

Choix pédagogiques

Pour déstabiliser et modifier les conceptions construites par la seule expérience perceptive, nous avons pris le parti dans un premier temps de limiter notre objectif à la construction d'une représentation opératoire du système informatique au travail. Ce niveau de base doit pouvoir suffire pour résoudre de façon raisonnée les problèmes posés par l'exécution des tâches proposées au cycle d'observation.

Ainsi il s'agit de faire découvrir les attributs essentiels qui permettent de comprendre le fonctionnement d'un système informatique et de repérer les relations qu'il y a entre ces attributs.

Nous avons retenu les attributs suivants :

- des données à modifier ;
- une instruction en cours d'exécution qui modifiera les données ;
- un « lieu » où l'instruction traitera les données : l'unité de traitement (microprocesseur) ;
- un « lieu » qui contient les instructions et les données activées et les résultats : la mémoire de travail (dénomination préférée à mémoire vive car elle est moins ambiguë et davantage porteuse de sens que cette dernière définition).

Modèle pédagogique

Nous avons choisi de tenter une adaptation du « modèle pédagogique visant l'abstraction », proposé par B. MARI-BARTH.

Ce projet consiste, à partir de la propre expérience de l'élève placé dans des situations provoquées, construites et choisies, à l'aider à mettre en place progressivement des images mentales structurées et efficaces.

Il s'agit donc de lui faire prendre du recul, de mettre en contradiction ses représentations premières et l'obliger à émettre des hypothèses. L'enseignant formule des questions éclaircissantes pour permettre aux élèves d'effectuer les opérations mentales de l'abstraction.

Dans cette expérience, il s'agit de faire dépasser le concret et le visible. La réflexion se fera d'abord dans des situations concrètes (l'ordinateur peut travailler ou ne peut pas travailler). Ces situations d'analyse seront d'abord des moments de l'action de l'utilisateur, elles seront ultérieurement fictives, donc plus théoriques.

D'autre part, il nous semble utile, pour aider à la construction d'une représentation mentale, de proposer aux apprenants de traduire schématiquement leurs conceptions.

Dispositif de l'expérimentation en cours de Technologie, niveau sixième.

Pour que les attributs essentiels que nous avons choisis de faire découvrir puissent être repérés, il est préférable d'éliminer au préalable des attributs plus concrets (éléments matériels et visibles du système) qui pourraient parasiter l'objectif de mise en évidence de la mémoire de travail. D'où la mise en place de la première phase du dispositif.

Phase 1

Consigne de travail à la maison :

Au collège nous utilisons des micro-ordinateurs. Où et pour quoi faire utilise-t-on des ordinateurs dans votre environnement ou dans les entreprises ?

En classe (Groupe-classe en salle banalisée)

Consigne : *Que faut-il à un ordinateur pour fonctionner ?*

Ecrire toutes les propositions au tableau.

Consigne : Classifier les propositions.

Il s'agit de faire regrouper dans une même classe tous les éléments matériels du Système Informatique (SI).

Consigne : *Classifier les éléments du SI en tenant compte de leur rôle.*

Cette classification est revue collectivement en reprenant chacun des éléments du SI, en cherchant s'il appartient à tous les SI recensés dans le travail maison et en montrant que les périphériques peuvent varier (exemple remplacer l'imprimante par la table traçante, supprimer la souris...).

Il s'agit d'arriver à faire regrouper les périphériques d'entrée, les périphériques de sortie et d'isoler comme constante : l'unité centrale de

traitement. Ce vocabulaire ne sera bien sûr donné qu'à la fin de la séquence (institutionnalisation).

Lors du premier contact avec l'ordinateur en début d'année, les élèves ont représenté dans leur cahier un système informatique sous forme d'un croquis rapide, assorti d'une légende.

Consigne : *le croquis que vous avez réalisé en début d'année était une représentation simplifiée d'un SI. Vous allez maintenant proposer un schéma qui montre ce que nous venons de mettre en évidence.* (Schéma influencé par le travail sur PC).

Le schéma est-il le même sur le MO5 ? (visite de la salle du Nano-réseau).

La confrontation et la comparaison des représentations permet au groupe d'arriver à une (des) représentation(s) possible(s) retenue(s) pour leur pouvoir explicatif, leur communicabilité, leur simplicité, donc la rapidité d'exécution.

La conclusion de cette séquence est de parvenir à un schéma du SI du type :

Entrée → Traitement → Sortie

Périphériques d'entrée → UC → périphériques de sortie

Cette schématisation est une première conceptualisation, il y a passage de la représentation physique du système à une représentation fonctionnelle.

Phase 2

Le dispositif conçu par B. MARI-BARTH, consiste à proposer des exemples : les exemples « OUI » et « NON » qui permettent à l'élève d'identifier progressivement la combinaison constante d'attributs essentiels par comparaison et opposition. Les attributs proposés sont validés ou invalidés au fur et à mesure de la présentation des exemples. Les propositions n'ont donc à aucun moment le statut d'erreur mais seulement celui de **propositions provisoires**.

Le professeur prépare matériellement les exemples sur les neufs postes de l'atelier bureautique et convie le groupe d'élèves à une analyse collective et successive de chacun des exemples.

Consigne orale : *Je vais vous présenter deux sortes d'exemples que j'appellerai les exemples « OUI » et les exemples « NON ». À chaque fois que vous serez en face d'un exemple « OUI », vous pouvez être sûr que l'unité centrale contient tous les éléments pour travailler. Votre tâche consiste à chercher tout ce que les exemples « OUI » ont en commun. Le premier exemple « NON » ne contient aucune des caractéristiques, ensuite vous allez*

trouver certaines caractéristiques présentes aussi du côté des « NON », mais jamais toutes à la fois. Dès que vous aurez une idée, je la marquerai au tableau. Si vous considérez qu'une proposition n'est plus valable, car non vérifiée par les exemples suivants, nous la rayerons du tableau. Sur une feuille vous écrirez vos propositions avant de nous les communiquer.

Premier exemple « OUI » :

TDT WORKS lancé - texte à l'écran -

Le professeur a sélectionné un mot et a activé la fonction mettre en gras.

Consigne : *Que faut-il à l'unité centrale pour travailler ?*

✧ Observation silencieuse. Temps de recherche personnelle. Propositions d'attributs écrits sur la feuille avant d'être énoncés...

Premier exemple « NON » :

TDT WORKS lancé - Texte à l'écran -

✧ Discussion ... Elimination, rejet d'attributs ou maintien, justification.

Deuxième exemple « OUI » :

Sketch lancé - Fonction rectangle du menu dessin activée -

La bande de communication indique le message : « Rectangle : Entrez un point ».

Exemple « OUI » :

TDT WORKS lancé - Texte à l'écran -

Le professeur a sélectionné un mot et a activé la fonction en gras.

Le moniteur a été ensuite éteint et pourra être remis sous tension.

Question du professeur : « *Qu'est-ce qui est nouveau dans cet exemple ?* »

Exemple « NON » :

TDT WORKS lancé - Disquette contenant les données introduites sous leurs yeux.

Le professeur a activé la fonction mettre en gras.

Une série d'exemples « OUI » et une série d'exemples « NON » :

Nous avons préparé une double série d'exemples (nouvelles inférences et nouvelles vérifications), mais ils ne sont pas présentés dans un ordre pré défini car ils seront choisis par l'enseignant dans son « stock » en fonction des attributs proposés. Il s'agit d'orienter la recherche en déstabilisant les élèves par des exemples « NON » dans lesquels figurent les attributs qu'ils considèrent comme essentiels et par des exemples

« OUI » dans lesquels ils ne figurent pas (comparaisons analytiques, rejet ou maintien, justification). Les derniers exemples permettront la généralisation.

Cette phase sera terminée quand les attributs et leurs relations auront été énoncés.

Les exemples devront permettre de découvrir ou confirmer

- qu'une information peut être présente au même moment dans deux éléments distincts du système ;
- qu'une information peut-être présente sous deux formes différentes (celle d'avant le traitement, celle d'après le traitement) au même moment dans deux éléments distincts du système ;
- qu'une information peut-être présente au même moment sous deux états : active ou en attente dans deux éléments distincts du système.

Phase 3

Cette phase doit permettre à tous les élèves de valider leurs représentations. Le professeur présente des exemples et les élèves doivent les identifier en tant qu'exemples « OUI » ou « NON » en argumentant leurs affirmations par l'énoncé des attributs.

Phase 4

Il est enfin demandé aux élèves d'inventer des exemples « OUI » ce qui permet une évaluation intermédiaire. (L'évaluation terminale sera menée dans une tâche nouvelle pour vérifier s'il y a changement par rapport aux comportements de départ et si la performance attendue est réalisée).

Phase 5

Nous proposons d'ajouter une phase au modèle de B.M. BARTH en raison de la particularité du thème de cet apprentissage qui se prête à une schématisation.

Consigne : *Complétez les schémas du SI avec les attributs que nous avons mis en évidence.*

Ces schémas devraient permettre de représenter les flux d'information.

Cette phase permet d'évaluer les acquis sur un plan formel.

Cette démarche est un exemple de situation d'apprentissage qui ne considère pas l'élève en temps que sujet passif recevant des connaissances sous forme de discours sans aucune assurance d'une modification de ses conceptions. Par contre, le type de méthode pédagogique que nous avons retenue déstabilise mais propose à chaque fois des éléments de reconstruction, ce qui permet à l'élève d'élaborer par lui-même son savoir.

Évaluation

Pour évaluer l'efficacité du détour méthodologique et de ses contenus, nous pensons que seule la réalisation d'une tâche peut être retenue pour vérifier que les connaissances sont opératoires.

Au cours du premier trimestre les tâches proposées ont été effectuées à l'aide du Traitement de texte.

Au deuxième trimestre, en Dessin Assisté par Ordinateur (DAO), les élèves ont été invités à utiliser une fiche technique vide enregistrée sur disque dur pour la compléter, puis l'enregistrer sous un nouveau nom sur la disquette.

Consigne : *La fiche technique enregistrée sur le disque dur sous le nom de fichier « fiche », sera utilisée pour réaliser toutes les fiches du dossier technique : Porte-crayon.*

- *Nomenclature*
- *Développé du corps*

Cette fiche type de départ devra toujours rester vierge sur le disque dur.

Chaque fiche du dossier sera sauvegardée sur disquette, sous un nom qui permettra d'en connaître le contenu.

Bien que guidée par des fiches sur les différentes fonctions, cette tâche est en général réalisée avec beaucoup d'erreurs lors de l'enregistrement (qui est fait par défaut sous le disque dur C:).

Conclusion

Nous n'avons pas obtenu des résultats très différents entre le groupe expérimental et le groupe témoin¹⁹ (10 %), mais par contre à la reprise du travail la semaine suivante, au cours de laquelle nous avons donné des consignes très directives :

- *allumer*
- *changer de répertoire : C: → A:*

¹⁹ Rappelons que le groupe témoin n'avait pas suivi la séquence « attributs » ; voir chapitre 5.

Les réactions des élèves devant la liste ne comportant pas leur fichier ont été totalement différentes dans les deux groupes : seuls les élèves du groupe expérimental ont été, dans leur majorité, capables d'analyser leurs erreurs et d'y remédier. Notre objectif a donc paru atteint. ■

Commentaires

Cette expérience est intéressante pour la dynamique qu'elle apporte dans le fonctionnement du groupe-classe à propos de notions nouvelles qu'il s'agit d'élaborer, souvent à partir d'idées fausses préexistantes chez les apprenants. La déstabilisation induite par les contre-exemples ouvre la voie aux corrections d'une manière qui semble efficace.

Le principal intérêt de cette séquence est d'avoir montré que l'on pouvait appliquer ce modèle aux situations visant à faire acquérir des savoir-faire, des opérations pratiques sur le dispositif. Cette démarche originale pourrait, à notre avis, être exploitée plus fréquemment.

UNE APPROCHE TECHNIQUE CENTRÉE SUR L'OBJET

L'équipe de Rouen, formée de mathématiciens, s'est intéressée à une approche des logiciels généraux (et principalement du traitement de texte, objet de nombreux stages) par la théorie des « langages orientés objet », support récent de langages de programmation de haut niveau. Nous voyons là un des rares exemples (mais l'avenir nous en apportera peut-être d'autres) de liaison étroite et fructueuse entre le niveau théorique de la « science informatique » et les utilisations des dispositifs micro-informatique sur le terrain.

²⁰ Modélisation et programmation objet

Il ne s'agit pas ici de faire un cours d'algorithmique ou de programmation, mais d'essayer d'analyser les raisons de l'évolution « objet » des logiciels, et surtout des conséquences induites pour l'utilisateur.

L'informatique n'effectue qu'un traitement formel des données qu'elle manipule. Or les données traitées sont, elles, porteuses de sens pour l'utilisateur. La difficulté pour le programmeur est donc de modéliser, dans une structure informatique aussi proche que possible des représentations générales hors informatique, les données dont il veut automatiser le traitement. C'est l'objectif principal d'une analyse en terme d'objets et les langages de programmation à objets sont le produit de cette réflexion.

Un objet, c'est un ensemble de *propriétés* (les propriétés ou « attributs » d'un caractère dans un document peuvent être la police, la

20. Contribution D. BERGUE, P. BRAS, O. FREMONT, P. JEANNE, C. SABOURIN.

taille...) et un ensemble de fonctions (**méthodes**) dont un sous-ensemble important sert à changer la valeur des propriétés.

L'orientation objet des logiciels les plus récents fait une large place à l'exploitation de ce schéma, avec pour l'utilisateur une simplification importante : à tout moment un sous-ensemble des données manipulées (caractère(s), paragraphe(s), cellule(s), document, logiciel, ...) est sélectionné (actif). On peut faire apparaître alors (en général par un clic sur le bouton droit de la souris) les fonctions et paramètres applicables à l'élément sélectionné (menus contextuels), ce qui évite à l'utilisateur la connaissance de la topographie des différents menus de commande. La première approche de cette réflexion par les concepteurs de logiciels a été l'utilisation de menus de commandes variables. Certaines commandes sont actives et d'autres inactives (grisées ou même absentes), en fonction du sous-ensemble des données qui est sélectionné. Cette situation est troublante pour un stagiaire, et à notre avis, seule une explication en terme d'objet ou fonction/argument (voir plus loin) est satisfaisante.

Exemple : notre analyse de la notion de fichier

Fichier : ensemble fini d'informations enregistrées sur disque.

Cette définition que l'on trouve sous des formes voisines dans la littérature est toutefois insuffisante.

Elle l'est au sens de l'informatique pure (mais ce n'est pas ici notre préoccupation), car elle ne permet pas de déterminer si un objet est un fichier ou non : le contenu d'une disquette répond à cette définition et n'est pas généralement un fichier.

Elle l'est pour l'utilisateur (non informaticien) car elle n'est pas opératoire : comment reconnaître un fichier ? Comment distinguer un fichier d'un autre ? Que puis-je faire avec un fichier ?

Pour affiner cette définition on peut tenter de considérer la notion de fichier comme une classe d'objets informatiques dont les représentants (instanciations) ont en commun d'avoir les mêmes propriétés, le terme propriétés regroupant les attributs (identificateur, taille...) et les comportements ou méthodes (dupliquer, renommer...).

Sans en établir une liste exhaustive, on peut tenter de pointer les attributs et méthodes les plus couramment mis en oeuvre lors d'une session de travail sur micro-ordinateur.

Un fichier possède	
attributs	Un identificateur : il regroupe ce que l'on nomme généralement le « nom » et le « chemin », et correspond à un emplacement physique sur dk.
	Un contenu : la suite d'octets le composant
	Une structure ou format : propriété qui fait qu'il peut être reconnu par tel ou tel logiciel.
	Une taille : volume de données mesuré en octets
	Une date de création
	Un certain nombre de valeurs logiques indiquant le statut d'un fichier : fichier caché, système...
	Un état : ouvert, fermé
	...
Un fichier peut être	
méthodes	créé
	supprimé
	dupliqué (avec changement d'identificateur)
	renommé (changement d'identificateur)
	modifié
	lu
	écrit
	...

On peut remarquer que l'effet des méthodes est *d'assigner des valeurs aux attributs*. Par exemple la création d'un fichier revient à attribuer une valeur à tous les attributs d'un fichier, le renommer en modifie certains...

Cette définition générique de fichier rassemble des éléments aussi divers que les fichiers programmes, les fichiers textes, les fichiers images etc. Tous ces éléments ont en commun les propriétés citées précédemment, et en particulier l'organisation logique et physique de leur contenu sur disque. Ce n'est que l'interprétation de leur format par le système d'exploitation (lui même composé de fichiers et de valeurs numériques) qui en fera un fichier programme ou un fichier de données.

Les fichiers programmes possèdent des propriétés supplémentaires, ce qui nécessite de les distinguer du type fichier générique.

Fichier programme : c'est un fichier qui possède la propriété de pouvoir s'exécuter.

Un fichier programme est d'abord un fichier. Il hérite donc l'ensemble des propriétés du type générique.

Il peut **s'exécuter** : c'est la faculté de pouvoir faire évoluer le système en passant par des états successifs²¹ ; cette propriété lui est propre et vient enrichir l'ensemble. Le rôle d'un programme est de modifier le système, seul ou en interaction avec d'autres programmes, avec ou sans intervention de l'utilisateur.

Remarque : une des modifications possible de l'état du système est la création ou la modification d'un fichier. Seul un programme est capable de réaliser ces tâches ; les méthodes d'un fichier utilisent des programmes.

Chaque instance de programme redéfinit la méthode « s'exécuter ». L'un traitera du texte, l'autre des données numériques... En fait la méthode globale « s'exécuter » est structurée en tâches élémentaires (sous-programmes : procédures ou fonctions) dont les plus usuelles sont sous-traitées à un programme spécial, **le système d'exploitation**.

Une session de travail avec un micro ordinateur utilise simultanément plusieurs fichiers, le système d'exploitation, un ou plusieurs logiciels d'application (fichiers programmes), un ou plusieurs fichiers de données.

Le schéma argument / fonction / paramètre

a) En général

Il nous semble que toute action d'un utilisateur d'un programme informatique peut se décrire en s'appuyant sur le schéma « fonction/arguments » ou encore « fonction/argument/[paramètres] », la suite des paramètres étant en nombre variable éventuellement vide. On peut remarquer que certaines situations utilisent des paramètres implicites (valeurs par défaut). Dans l'exemple 4, le fichier sera enregistré avec son identificateur actuel.

L'effet d'une fonction est la modification de l'état d'un ou plusieurs éléments constituant le « système informatique », c'est à dire les mémoires.

L'effet est tout à fait comparable à celui d'un programme : une fonction est d'ailleurs souvent appelée « sous-programme », une session informatique étant alors une suite (composition) de fonctions. Le résultat du travail de l'utilisateur, c'est à dire l'état du système à la fin de la session, dépend alors du choix et de l'ordre des fonctions utilisées.

21. État du système : contenu, à l'instant t, de la mémoire vive, de la mémoire écran, des disques...

	fonction	argument	paramètres
1)	formater	la disquette située dans le lecteur A :	au format 720 Ko
2)	activer la commande	quitter du menu fichier	
3)	enregistrer	le document de travail	sous le nom FIGURE1.G2D
4)	enregistrer	le document de travail	
5)	transformer	les caractères sélectionnés	en gras, italique et taille 12

La composée de plusieurs fonctions étant elle même une fonction, une session de travail est donc la construction par l'utilisateur d'une fonction complexe, transformant le système de l'état initial, avant l'allumage de la machine, à l'état final après extinction. (Certains langages de programmation, comme LISP par exemple, assimilent un programme à une fonction).

Nous avons cherché à exhiber un contre-exemple à ce schéma, sans y parvenir. Puis nous avons essayé d'analyser ce qu'est une action de l'utilisateur. À notre avis elle peut se représenter de la manière suivante :

verbe (l'action)	un complément d'objet direct ou indirect	autres compléments
fonction	argument	paramètres éventuels

Les questions les plus intéressantes sont toutefois relatives à l'intérêt et aux difficultés que représente ce schéma pour l'utilisateur. Cette représentation a l'avantage d'être générale, car elle s'applique aussi bien à des objets physiques tels que les disques mais aussi des objets abstraits comme les répertoires, les fichiers, ou n'importe quel type de données en cours de traitement (voir développement sur le traitement de texte). Cependant ce n'est sûrement pas la seule façon de schématiser un travail sur ordinateur.

b) Schéma Argument/Fonction appliqué à la notion de fichier

L'exposé fait ici est lié à la notion de fichier, mais il peut être transposé aux manipulations relatives aux autres objets physiques ou abstraits cités plus haut.

Le choix du sous-ensemble des attributs et méthodes exposé est le fruit des observations de la première phase de notre travail. Nous avons séparé très nettement la description des propriétés des fichiers (attributs et méthodes) de leur mise en oeuvre. En effet ces propriétés sont générales alors que leurs manipulations dépendent du contexte logiciel. L'opération de duplication d'un fichier est toujours la même, mais la

façon de procéder est fonction de l'interface, ligne de commande, gestionnaire de fichiers...

Le schéma argument/fonction/paramètres est dans cet exemple bien présent. La fonction de copie s'applique à un ou plusieurs fichier(s) désignés avec comme paramètre optionnel l'identificateur du ou des fichiers cibles (le paramètre est en fait obligatoire, mais s'il n'est pas ou que partiellement signalé, c'est une valeur totalement ou partiellement implicite qui sera prise). Une difficulté supplémentaire apparaît ici et qu'il convient de signaler car elle se présente dans divers contextes. Suivant l'interface choisie, l'ordre de désignation de la fonction et de ses arguments est inversé : en ligne de commande on écrira « COPY MONFIC.EXT ... » (fonction *puis* argument), alors que dans une interface graphique, on sélectionnera *d'abord* les fichiers à copier *puis* la commande de copie, que cela se fasse à l'aide d'un menu ou par une opération « glisser / déposer » à l'aide de la souris.

Proposition pour une généralisation : exemple du traitement de texte

La présentation de type argument/fonction exposée plus haut est-elle transférable pour une formation à un logiciel donné ? Nous nous sommes penchés sur l'enseignement du traitement de texte. Différents logiciels de traitement de texte ont servi de support.

Les objectifs de la formation servant de test était la suivante :

- amener en un minimum de temps de formation (moins de 6 heures) des utilisateurs ayant suivi notre présentation des notions de flux d'informations et de gestion des mémoires, mais entièrement novices pour l'utilisation du traitement de texte, à maîtriser les fonctionnalités de base (qui sont des fonctionnalités communes à de nombreux traitement de texte).

fonctionnalités de base : ouvrir un document (éventuellement vide), en modifier le contenu, mettre en forme caractères et paragraphes, utiliser le COPIER/COUPER-COLLER, éventuellement effectuer une vérification orthographique, sauvegarder le nouveau document, se préparer à l'impression, et imprimer.

- être capable de réaliser, en autonomie, une séquence équivalente avec un logiciel de traitement de texte différent.

Pour obtenir une présentation de type argument/fonction, une analyse des données manipulées par un traitement de texte et surtout leur structuration en différents types d'éléments est nécessaire.

Un texte :

- c'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 (cette structure n'est pas présente dans tout traitement de texte et nous l'avons pas prise en compte dans les stages) ;
- c'est un document.

Le terme « suite » a été préféré au terme « ensemble », pour une raison mathématique liée à l'ordre et à l'unicité : en mathématique les ensembles formés des lettres des mots « aval », « lava » et « val » sont égaux à l'ensemble $\{a,l,v\}$, alors que pour le contenu d'un document, il y a une différence. De même, nous utiliserons le terme de sous-suite.

Les données manipulées peuvent présenter un niveau de structuration supplémentaire si plusieurs documents sont ouverts en parallèle.

À tout moment une sous-suite (éventuellement vide lorsqu'il s'agit de caractères) de chacun des éléments de structure est active (sélectionné) de manière implicite ou par une action déterminée de l'utilisateur. Si aucune sélection n'a été volontairement réalisée par l'utilisateur, le document, la page, le paragraphe actifs sont déterminés par l'emplacement du curseur. Dans tout ce document le terme **sélection** désignera l'ensemble formé par les sous-suites actives de chaque type.

Chaque type d'élément possède un ensemble propre d'attributs et de fonctions pour changer leur valeur (chaque attribut possède une valeur, éventuellement implicite lors de sa création, prise dans un domaine de valeurs possibles propre à chaque attribut). On peut d'ailleurs se servir de cette analyse pour définir un type, ce qui permettra de donner un sens dans un traitement de texte aux mots « caractère », « paragraphe »... et qui peut différer de leur sens hors contexte informatique.

Deux éléments sont du même type s'ils ont les mêmes attributs et fonctions. Ce qui implique, par parenthèse, que le type caractère, par exemple, peut différer d'un logiciel à l'autre, d'où il se pose plus généralement des problèmes de transfert de données entre logiciels.

L'action d'un utilisateur est donc de connaître le type d'objet dont il veut modifier un ou plusieurs attributs, d'opérer une sélection si besoin et de lui appliquer la fonction appropriée, éventuellement accompagnée d'un choix de paramètres (argument/fonction/[paramètres]).

Exemple : définir les retraits d'un paragraphe

Sélectionner le paragraphe : suivant le contexte logiciel, il suffira de placer le curseur dans le paragraphe en question, ou d'opérer une

sélection plus explicite. Si le traitement doit s'appliquer à plusieurs paragraphes simultanément, il faudra une sélection explicite. Les logiciels de traitement de texte ne permettent pas en général de sélection discontinue, mais cette limitation n'est que technique, car rien, au plan opératoire ne s'y opposerait ; c'est par ailleurs une limitation découverte rapidement et regrettée par les utilisateurs que nous avons observés.

Désigner la fonction appropriée : le procédé offert par les logiciels est d'aller choisir dans un menu la commande correspondante. L'uniformisation des barres de menu rend la localisation plus aisée, pour le transfert des savoir-faire vers un autre logiciel de traitement de texte et même vers un autre type de logiciel. Les types d'éléments « caractère », « paragraphe »... sont présents dans les intitulés de menu ou commande, ce qui rend le choix de la commande d'autant plus aisé que l'on sait quel est l'élément dont on souhaite modifier les attributs. Les logiciels les plus récents, de tous types, proposent des menus contextuels. Les commandes correspondant à des fonctions non applicables à la sélection sont grisées. La tendance actuelle va plus loin dans ce sens : un clic sur le bouton droit de la souris fait apparaître dans un menu supplémentaire une liste des commandes les plus courantes et applicables à la sélection.

La fonction étant choisie, il faut en **régler les paramètres éventuels**. On peut remarquer que la plupart des fonctions de mise en forme permettent le choix de la valeur de plusieurs attributs simultanément. C'est un choix de conception du programmeur, qui s'est déterminé pour des fonctions moins nombreuses mais plus complexes (c'est à dire dont les paramètres sont en nombre plus important). A ce moment, les attributs sont désignés par leur nom : retraits, espacement, interlignage, alignement. On ne peut faire l'économie de la définition de ces termes. Certains sont clairs, car leur sens dans ce contexte est le sens courant. D'autres sont moins simples ou liés à d'autres attributs : un retrait est un espacement par rapport à une marge. Un attribut prend ses valeurs dans un domaine pré-défini : ainsi certaines valeurs peuvent être refusées.

Proposition pour une séance d'enseignement de trois heures

Avant tout travail individuel sur micro ordinateur, quelques minutes d'explications collectives vont permettre de préciser le fonctionnement global d'un traitement de texte et les différents types d'éléments qu'il manipule : un traitement de texte traite du texte ! Qu'est-ce qu'un texte ? (voir plus haut). C'est l'occasion de préciser le sens de certains termes (marges, retraits...).

- Une séquence collective courte (15 minutes) de saisie de quelques lignes de texte :
 - passage à la ligne automatique, création d'un nouveau paragraphe, fusion de 2 paragraphes, navigation dans un texte

à l'aide des touches de déplacement, manipulations permettant la sélection, correction et insertion de texte) ;

- quelques commandes de sécurisation : ÉDITION-ANNULER et la touche d'échappement.

Une des particularités d'un public d'adultes est la réticence à essayer une action dont on est pas sûr de l'effet, alors qu'un enfant n'hésite pas à fonctionner par essais/erreurs. Or cette manière de procéder est un moyen (peut-être peu rationnel, mais qui aura le courage de lire 500 pages de documentation avant de commencer ?) d'auto-apprentissage des fonctionnalités d'un logiciel. Il est donc important de faire passer l'idée que rien n'est définitif et que l'on peut à tout moment (même si ce n'est pas tout à fait exact) revenir en arrière et gommer une erreur.

- Travail individuel sur micro-ordinateur : le stagiaire ouvrira un texte saisi à l'avance par le formateur et devra y opérer un certain nombre de transformations et mises en forme afin d'en faire un document imprimable. Chaque stagiaire a une feuille de consignes lui indiquant les manipulations qu'il doit réaliser. Cette feuille de consigne a un double but :
 - donner une autonomie à chacun. L'expérience prouve qu'il est presque impossible de récupérer l'attention collective d'un public travaillant sur micro-ordinateur. De plus, l'intervention du formateur n'aura que peu de chances d'être pertinente du fait des différences de rythme de travail de chacun ;
 - les consignes sont données de sorte que les premières soient très directives (les actions y sont décrites entièrement et étape par étape) et qu'au fur et à mesure de l'avancement du travail, le stagiaire ait davantage d'initiatives à prendre.

Expérimentations et analyse

Cette séquence de trois heures a été mise en place dans de nombreux stages (et pas uniquement pour des professeurs de mathématique), dont l'objectif immédiat n'était pas l'apprentissage approfondi d'un traitement de texte particulier (un tel objectif est-il par ailleurs vraiment nécessaire pour des utilisateurs débutants ?). Plusieurs logiciels ont été utilisés.

La plupart des stagiaires avaient déjà reçu un enseignement basé sur le même schéma présentant les flux d'information et la gestion des mémoires. Lorsque cette présentation du traitement de texte a été proposée lors d'une première séance à des stagiaires totalement débutants, la séquence a malgré tout fonctionné, mais a pris plus de temps. Les problèmes rencontrés dans ce cas n'ont pas été d'ordre conceptuel, mais plutôt fonctionnels : prise en main du clavier et de la souris, repérage des informations affichée à l'écran...

Cet enseignement n'a pas la prétention de former des professionnels de l'édition. Il a par contre celle de permettre une autonomie dans des tâches de base, de donner une vision globale de ce qu'est un traitement de texte pour permettre au formé de transférer ces connaissances dans un autre contexte logiciel ou une autre évolution du même logiciel. Un autre objectif visé est d'avoir donné au stagiaire les moyens de progresser de manière individuelle dans ses pratiques sur son traitement de texte personnel, qui bien souvent n'est pas celui qui a servi de support à la formation qu'il a reçue.

L'objectif était au départ de concevoir une formation de 2 journées au plus. La séquence décrite occupe 3 heures. En fait quelques compléments ont parfois été apportés, en particulier pour des professeurs d'une discipline scientifique, dont les documents intègrent des formules mathématiques, des tableaux et des graphiques. Une deuxième demi-journée a été consacrée dans ce cas à l'apport de ces compléments, et a servi en outre au réinvestissement des connaissances acquises lors de la première demi-journée.

En guise de conclusion

Cette méthode d'enseignement et son contenu n'ont été testés que dans le contexte très particulier de formations s'adressant à des enseignants, de mathématiques pour la plupart. Il est donc difficile d'en tirer des conclusions générales. De plus, pour les formations mises en place cette année, le recul permettant de juger de leur efficacité manque. Les grilles d'observation, remplies au terme de la formation, sont peu significatives, les stagiaires n'ayant pas eu suffisamment de temps pour réinvestir l'enseignement reçu.

Par contre, nous avons configuré, il y a un an, dans plusieurs sites (établissements scolaires ou chez des collègues) un micro-ordinateur démarrant sur un menu textuel ou graphique permettant de choisir les logiciels présents. Nous avons pu suivre des collègues qui avaient reçu une formation l'année dernière, alors que le contenu était en cours d'élaboration. Ces micro-ordinateurs sont utilisés régulièrement, pas au maximum de leurs possibilités, par une partie des enseignants ayant suivi la formation, sans qu'il n'ait été nécessaire d'intervenir. Les éventuelles demandes qui surviennent sont d'ordre pédagogique ou d'approfondissement d'un logiciel.

Objets manipulés dans le traitement de texte de WORKS pour WINDOWS

Objet	Attributs	Actions	Lisibilité	Comment ?
Fichier	<ul style="list-style-type: none"> - Nom - Taille - Ouvert ou fermé - Etc. 	<ul style="list-style-type: none"> - Renommer - Fermeture - Ouverture - Création 	<ul style="list-style-type: none"> - Surveiller le nom de la fenêtre active 	<ul style="list-style-type: none"> - Se placer dans le document concerné - Fichier
Document	<ul style="list-style-type: none"> - Longueur feuille - Largeur feuille - Marges - N° de la 1ère page - Contenu - Etc. 	<ul style="list-style-type: none"> - Changer les valeurs par défaut (marges...) - Correcteurs et synonymes - Rechercher, remplacer - Déplacer, dupliquer, supprimer des blocs de texte - Saisie du texte - Insertion (saut de page) 	<ul style="list-style-type: none"> - Surveiller les crochets de la règle - Commentaires à l'écran 	<ul style="list-style-type: none"> * Pour mettre en page : - Se placer dans le document concerné - Mise en page * Pour modifier le contenu - Se placer où l'on veut insérer ou sélectionner le bloc de texte concerné - Edition
Paragraphe	<ul style="list-style-type: none"> - Alignement - Interligne - Espacement - Retraits - Encadrement - Tabulations - Etc. 	<ul style="list-style-type: none"> - Changer les valeurs par défaut 	<ul style="list-style-type: none"> - Commentaires à l'écran et Aperçu avant impression 	<ul style="list-style-type: none"> - Se placer sur le paragraphe concerné ou sélectionner le(s) paragraphe(s) concerné(s) - Format
Caractères	<ul style="list-style-type: none"> - Police - Taille - Position - Style - Etc. 	<ul style="list-style-type: none"> - Changer les valeurs par défaut 	<ul style="list-style-type: none"> - Commentaires à l'écran et Aperçu avant impression 	<ul style="list-style-type: none"> - Sélectionner le(s) caractère(s) concernés(s) - Format
Objet externe	<ul style="list-style-type: none"> - Contenu - Taille 	<ul style="list-style-type: none"> - Modifier le contenu de l'objet, sa taille 	<ul style="list-style-type: none"> - Un seul clic sur l'objet suffit pour le sélectionner 	<ul style="list-style-type: none"> *Pour modifier le contenu : - Sélectionner l'objet - Edition Editer l'objet * Pour modifier la taille : - Sélectionner l'objet - Format

Objets manipulés dans le gestionnaire de Fichiers de WINDOWS

Objet	Attributs	Actions	Lisibilité	Commentaire
Fichier	- Identificateur - Taille Etc.	- Créer - Renommer - Supprimer - Déplacer - Modifier le contenu	- Commentaires à l'écran (nouveau nom, changement de répertoire, de taille etc.)	- Se placer sur le fichier ou sélectionner les fichiers concerné(s) - Choisir la commande appropriée dans le menu Fichier
Répertoire	- Nom Etc.	- Créer - Renommer - Supprimer - Déplacer	- Nouvelle arborescence	- Se placer sur le répertoire concerné - Choisir la commande appropriée dans le menu Fichier
Disque	- Capacité (Taille, Densité) Etc.	- Formater - Afficher le contenu - Copier	- Message si la disquette est illisible - Le témoin lumineux du lecteur s'allume	- Activer le lecteur concerné (si besoin) - Choisir la commande appropriée dans le menu Disque

