

CHAPITRE 4

LES ÉLÉMENTS LOGICIELS

La mise à disposition, l'utilisation et la conservation sur différents supports des données et des programmes nécessitent des connaissances et des savoir-faire dont l'acquisition n'est ni aisée ni immédiate.

La gestion globale du système informatique va se présenter comme un ensemble de problèmes à traiter à la fois globalement et en détails. Globalement, parce que nous allons avoir affaire à des notions très générales fédérant toutes les opérations : les mémoires, les fichiers, les dialogues opérateur-système... En détails, parce que nous devons nous pencher précisément sur la concrétisation de ces questions pour chaque application de l'ordinateur dans son cadre disciplinaire.

Après avoir ébauché une classification des niveaux d'utilisation de l'ordinateur, nous proposons une synthèse des observations des terrains concernant le système d'exploitation et les fichiers. Mais nous nous sommes bien vite aperçus qu'on ne pouvait évoquer les fichiers sans parler de la notion de mémoire. Ce concept fondamental fera à lui seul l'objet du chapitre suivant.

1 Niveaux d'utilisation et niveaux de compétence

Le sujet de notre étude étant *l'utilisation de l'ordinateur en tant qu'outil, quel que soit le domaine d'application*, il nous a paru nécessaire de définir un cadre permettant de donner un sens à ces termes.

Le mot utilisation peut recouvrir des séquences de travail diverses, dont les objectifs peuvent mettre en oeuvre des savoirs et savoir-faire variés, et de niveaux très différents.

Le terme « maîtrise » est sans doute plus clair : il s'agit pour l'utilisateur d'être capable de réinvestir, en autonomie, les connaissances acquises, pour réaliser une séquence du même type, éventuellement dans un environnement différent (transfert).

Notre premier travail a donc été de tenter de classifier les différents types d'utilisation, en fonction de l'objectif de la séquence. Nous avons

1 Contribution équipe de Rouen : D. BERGUE, P. BRAS, O. FREMONT, P. JEANNE, C. SABOURIN.

essayé d'établir une hiérarchie dans les séquences à effectuer. Nous entendons par hiérarchie que l'ensemble des savoirs et savoir faire nécessaires à un niveau soit strictement inclus dans celui du niveau suivant.

Niveau a : Séquence dont l'objectif n'est pas la production ou la modification de données

Cette rubrique recouvre l'utilisation des logiciels dits « fermés », tels certains logiciels d'EAO ou de recherche documentaire (c'est du même ordre que l'utilisation d'un service télématique).

On peut schématiser la séquence de l'utilisateur de la façon suivante :

- allumer la machine
- activer le logiciel (par son identificateur, ou à l'aide d'un menu)
- naviguer dans les menus du logiciel
- répondre à des questions ou des choix d'options
- quitter le logiciel
- éteindre la machine

Remarque : lors d'une utilisation de ce type, il peut y avoir production de données (par exemple l'évaluation de l'élève pour son travail, mais ce n'est pas une action délibérée de l'utilisateur).

Niveau b : Séquence utilisant ou produisant des données modifiables

Ce titre recouvre l'ensemble des séquences les plus fréquentes en milieu scolaire. Cela va de la production (ou modification) d'un texte, à la PAO, de la réalisation d'une figure de géométrie à la programmation. La différence essentielle avec le type précédent est que l'utilisateur va ici devoir désigner les données à « ouvrir » et à « sauvegarder ».

Cette classe est relativement vaste et nous a conduits à une distinction plus fine en fonction de la séquence qu'entend réaliser l'utilisateur.

Niveau b1 : Un logiciel, un type de document

Le logiciel utilisé est le seul activé et il gère un document (éventuellement plusieurs du même type entre lesquels il est possible d'échanger des données) ;

Niveau b2 : Production d'un document composite

Le document traité contient des données produites par différents logiciels. C'est le cas lors de l'utilisation d'un « intégré » ou d'un travail avec un environnement permettant l'échange de données entre applications. La distinction avec le type précédent vient de problèmes possibles pour la reconnaissance du format des données échangées, et de la mise en oeuvre plus ou moins transparente de plusieurs logiciels.

Niveau b3 : Archivage ou transport de données d'un support à un autre

Faire une copie de sauvegarde, produire des données avec une machine et les utiliser avec une autre.

Remarques :

- Il est probable qu'une session de travail sur ordinateur puisse faire appel à plusieurs type de séquences. Est-il raisonnable de produire un document sans en réaliser une copie de sauvegarde ?
- Bien que situé au niveau le plus haut de cette hiérarchie, le niveau b3) ne doit pas être pris pour un niveau maximal. Il est des compétences de nature différente qu'un utilisateur de niveau b3) ne possède pas. Dans la plupart des observations (évidemment chez les élèves mais aussi chez les stagiaires débutants) nous nous sommes limités à une utilisation d'un micro ordinateur « prêt à l'emploi » (ordinateur convenablement configuré). Or, ce micro ordinateur a dû être préparé. Des compétences différentes et/ou supplémentaires sont assurément indispensables pour cela. Ce domaine de compétences a été écarté de notre étude (cela reviendrait à observer seulement des personnes ayant atteint un niveau de connaissances assez élevé) , mais il mériterait certainement une réflexion spécifique. En effet, ces compétences sont nécessaires à un enseignant chargé, dans le cadre des programmes, de faire travailler les élèves sur un micro ordinateur. De plus, l'ordinateur familial n'est plus une exception. Des connaissances plus vastes doivent être mises en oeuvre pour apporter une réponse aux questions du type :

Quel ordinateur et quels logiciels dois-je me procurer, en fonction de mes objectifs d'utilisation actuels et futurs ? Comment rendre fonctionnel le matériel acheté ? Comment obtenir du matériel le maximum d'efficacité ? Quels éléments puis-je ajouter pour le faire évoluer ? ■

En fait, les réponses à ce genre de questions sont actuellement apportées partiellement par les formateurs aux enseignants déjà bien familiarisés avec l'informatique.

On pourrait se fixer comme but, dans l'avenir, qu'une formation généralisée dans une rationalité bien comprise permette à tout enseignant, moyennant un travail raisonnable (et non, comme cela se pratique actuellement, dans un investissement intellectuel colossal et pas toujours efficace), de joindre ces éléments aux autres critères de choix (institutionnels, budgétaires, etc.) pour installer un équipement informatique performant et à la portée du plus grand nombre d'utilisateurs.

LES PRINCIPAUX PROBLÈMES POSÉS PAR LE SYSTÈME D'EXPLOITATION

Le système d'exploitation est le premier élément logiciel rencontré après la mise sous tension de l'installation. Pour un débutant, cela signifie qu'avant d'entrer concrètement dans l'activité promise par le formateur (par exemple la création d'un document en traitement de texte), il doit réaliser un certain nombre d'opérations sans se tromper et, au départ, sans en connaître le sens précis. Dans la mesure où il reçoit des explications qui vont l'inciter progressivement à faire des essais mais aussi l'amener à commettre des erreurs inévitables, l'entrée dans l'univers du logiciel peut être ressentie comme un épisode pénible, voire décevant et décourageant pour la suite (c'est un souvenir que l'on a souvent entendu évoquer par des personnes expérimentées à propos de leurs débuts). A ce propos, on remarque dans le dépouillement des questionnaires à tous les niveaux collège que le Système d'Exploitation a tendance à être nié : c'est *le programme* (traitement de texte) *qui démarre* (observations de l'équipe de Toulouse).

Comment aborde-t-on le système d'exploitation ?

Les gestionnaires de système ont considérablement évolué ces dernières années, tant du point de vue de leurs performances que de celui de leur diffusion. Les installations Macintosh n'ayant pas été proposées dans le système éducatif jusqu'à une date très récente, les interfaces graphiques de gestion (basés essentiellement sur les icônes et la souris) sont apparus il y a peu de temps avec WINDOWS. Auparavant, les utilisateurs avaient suivi avec peine la progression des DOS, se heurtant sans cesse aux contraintes du langage formel de dialogue et à l'absence d'aide « en ligne » cruellement ressentie (« nom de commande ou de fichier incorrect »...).

Le travail sous DOS : un mauvais souvenir ?

Lorsque nous avons commencé cette étude (1991), la plupart des installations fonctionnaient sous DOS. Une des grandes questions qui nous soient immédiatement parvenues a été celle des difficultés de travail avec ce langage (très) formel, peu tolérant aux fautes. Effectivement, tous les utilisateurs se plaignaient de la lourdeur des dialogues et commettaient beaucoup d'erreurs à ce niveau, tant du point de vue strictement syntaxique que du point de vue sémantique, la plupart du temps par ignorance de la signification précise des commandes et des messages.

☞² Des observations effectuées en quatrième, troisième, seconde, première et en stage MAFPEN apportent les exemples d'erreurs suivants :

Messages ignorés, non lus : blocages pouvant s'éterniser (ex : « Date/Heure »), mal compris ;

Vocabulaire inconnu, codé, analogies trompeuses ;

Syntaxe : ambiguïté des rapports entre la syntaxe informatique et la grammaire de phrase traditionnelle

Même logique apparente : COPY A:ESSAI.DOC B:
(place/fonction des termes)

Ellipse : COPY ESSAI.DOC B:

Pièges : « impression en cours : tapez ESC pour interrompre ». ■

Nous ne nous étendrons pas sur ce niveau du problème, suffisamment connu. Citons simplement un chiffre provenant d'une observation effectuée sur des stagiaires MAFPEN par l'équipe de Rouen (travail sur WORKS) :

Parmi des erreurs importantes en nombre et en genre, on relève plus de 50% d'échec au lancement de DOS ou de l'application lors de la quatrième séance !

On retrouvera la mention d'erreurs syntaxiques dans tous les sites d'observation.

La présentation WINDOWS

La diffusion rapide de l'environnement graphique d'exploitation WINDOWS, inspiré des outils qui ont assuré le succès du MACINTOSH, a certainement contribué à libérer l'utilisateur des contraintes les plus lourdes de la micro-informatique. Remplacer la communication clavier-écran enserrée dans une syntaxe rigide par une présentation graphique où il suffit de pointer et cliquer une icône avec la souris relève du miracle et renvoie encore une fois à la dimension mythique de cette machine magique.

Cependant, les enseignants ont rapidement constaté que l'usage de cet outil ne supprimait pas toutes les difficultés. Passé les premiers enthousiasmes, les erreurs ont commencé à surgir. En fait, une catégorie de difficultés le plus souvent rencontrées se situe au niveau qu'on peut considérer à la fois comme le plus banal et le plus puissant de l'outil : le multi-fenêtrage. Ouvrir plusieurs fenêtres, travailler dans chacune d'elle et passer de l'une à l'autre font mettre en oeuvre implicitement un

2 Equipe de Toulouse : A. M. LACANAL, D. MARTY, D. ROSSO, M. SAINT-MARTIN.

raisonnement par analogie (voir ci-dessous) qui fait assimiler les fenêtres à des feuilles de papier superposées (cette analogie était évidemment intentionnelle de la part des concepteurs du logiciel). Mais pour l'utilisateur, la notion d'activation de fenêtre n'a pas vraiment son équivalent dans le domaine matériel connu. De la même manière, les opérations sur les icônes représentant les applications ne peuvent pas être directement assimilées à d'autres opérations du domaine source de l'analogie. Toute la gestion des fichiers de données et de programmes réalisée en amont de ces manipulations graphiques n'apparaît pas à l'utilisateur de façon évidente ; il semblerait cependant indispensable qu'il puisse en comprendre certains éléments.

☞³ Un enseignant a analysé le rôle de WINDOWS à la fois comme *interface* (intermédiaire) et comme *cadre intégrateur* (« agent travaillant pour son propre compte ») :

« Dans le premier, on trouve essentiellement un outil de convivialité facilitant les opérations du DOS en matière de gestion de fichiers et d'applications. Dans le second sont traités tous les échanges de données entre applications, impossibles au seul premier niveau, mais qui les englobent. On remarque que dans une telle situation, les limites d'un apprentissage avec comme outil une documentation de type « guide pas à pas » apparaissent clairement, car on ne peut prévoir ni décrire toutes les situations imaginables, **et le problème posé est bien celui d'une conceptualisation des principes de fonctionnement nécessaires aux utilisations, même les plus courantes.** » ■

Cette opinion nous paraît très importante, car d'une manière générale, les outils de type WINDOWS (et a fortiori les systèmes MACINTOSH) sont promus par des arguments mettant en avant leurs facilités de mise en oeuvre sans nécessité de compréhension profonde, ce qui, à notre sens, est tout à fait faux. A contrario, les notices des concepteurs de logiciels présentent généralement une information exhaustive car ne pouvant pas décrire toutes les situations d'utilisation, comme le fait justement remarquer notre collègue ; cela rend ces aides difficiles d'accès - et pour le moins rebutantes - à un utilisateur débutant.

Notre collègue complète son approche de WINDOWS par des remarques générales ou de détails :

☞³• le rapprochement apporté par WINDOWS entre la perception du document à l'écran et sa réalité (future) sur papier donne un avantage (le réalisme des polices et de la mise en page) dispensant souvent d'essais d'impression, mais peut entraîner des confusions entre objets (écran, mémoire et papier) ;

3 D. ROSSO, Toulouse.

- 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 de représentation des objets manipulés par l'informatique, sans les supprimer.

Nous notons encore l'induction par le système de confusions durables entre opérations sur fenêtres et opérations sur fichiers, et nous nous souvenons d'avoir déjà constaté (lors d'un travail réalisé en 1985) que « seul ce qui se voit existe... tout ce qui se voit est réel ». ■

Point théorique sur le raisonnement par analogie et l'anthropomorphisme

Parmi les différents raisonnements utilisés par le sujet pour apprendre et pour fonctionner, le raisonnement par analogie occupe une place importante (en général) et tout particulièrement en informatique. Il consiste à établir une relation entre une situation (ou un problème) connu (source) qu'on juge analogue à la situation inconnue, au problème qu'on doit résoudre (cible).

Il comporte trois phases de mise en œuvre :

- l'évocation, c'est-à-dire la recherche de la situation source,
- l'appariement, mise en correspondance des éléments analogues entre la source et la cible,
- l'évaluation de la pertinence de l'analogie.

On relève trois principales caractéristiques du raisonnement par analogie :

- les caractéristiques de surface : tout ce qui « entoure » le problème,
- les caractéristiques structurelles, qui sont liées à la structure profonde du problème,
- les caractéristiques pragmatiques, liées au but du raisonnement ; elles conditionnent souvent la réussite de la résolution du problème.

Une hypothèse qui a retenu notre attention⁴ est la suivante : le transfert (la réalisation du raisonnement par analogie) serait facilité quand les sujets ont élaboré des solutions à un certain niveau d'abstraction. Or ce qui pourrait faire défaut aux utilisateurs non experts dans nos situations, c'est justement le processus d'abstraction, ou plus précisément les capacités pour réaliser des abstractions face au problème posé, dans une situation donnée.

L'anthropomorphisme est une particularisation du raisonnement par analogie ; il consiste à identifier le système informatique à une personne humaine, dans toutes ses capacités de raisonnement et d'action, ce qui est évidemment abusif.

⁴ Lire : CAUZINILLE-MARMÉCHE, E., MATHIEU, J. et WEIL-BARAIS, A. [1983]. *Les savants en herbe*. Berne (Suisse), Peter Lang.

Reprenons l'exemple de l'analogie « papier » des environnements graphiques de type WINDOWS, qui devrait « passer » facilement. Dans la situation source (le papier), les opérations sont concrètes sur des objets réels : manipuler des feuilles de papier. Dans la situation cible, il y a plusieurs niveaux d'abstraction, sur les objets et les opérations :

- les objets sur lesquels on travaille sont des images écran, le concret n'est que la partie affichage du tube-image,
- ces images sont déjà des représentations de textes mémorisés dans des fichiers (il existe donc un double niveau d'abstraction).

Les opérations consistent principalement en :

- 1) faire apparaître un texte : CHARGER-OUVRIR,
- 2) intervertir les feuilles : c'est en cliquant sur la feuille sur laquelle on veut travailler qu'on la fait venir au premier plan de l'écran ; elle peut avoir complètement disparu, il faut alors activer le mot « écran » pour la faire réapparaître et surtout se souvenir qu'elle existe et qu'elle est encore disponible malgré sa disparition.

Ce sont des opérations abstraites, qui ont peu de rapport avec la manipulation physique de feuilles de papier.

Autre exemple : la hiérarchie dossier/document : pourquoi ce principe si simple et si utilisé dans le domaine « papier » fonctionne-t-il mal dans une structure informatisée « analogue » ?

Nous proposons l'hypothèse suivante : les difficultés proviendraient des abstractions nécessaires *tant au niveau des objets qu'au niveau des opérations*. Nous reprendrons cette analyse plus loin.

Les méfaits de l'anthropomorphisme

Outre les habituels dialogues très réalistes avec la machine : « il (l'ordinateur) m'a fait... », un abus par anthropomorphisme nous est montré dans le cadre de l'enseignement de la comptabilité en lycée professionnel (Toulouse). L'enseignante⁵ remarque qu'il existe des confusions entre le rôle de l'ordinateur chargé uniquement de tester l'égalité d'une écriture comptable et le rôle de l'utilisateur qui est de saisir les écritures justes. Dans l'esprit de l'élève, dès que l'ordinateur a accepté l'écriture en raison de son égalité comptable, ***celle-ci est forcément exacte*** ; l'ordinateur semble ainsi dispenser l'élève du moindre contrôle de vraisemblance, ce dernier attribuant au premier une véritable capacité de réflexion comptable. On imagine facilement les conséquences de telles croyances...

5 A. M. LACANAL.

Les menus déroulants

Les menus déroulants sont évidemment plus conviviaux que la traditionnelle entrée des commandes au clavier, mais ils induisent des difficultés nouvelles et spécifiques concernant l'accès aux opérations, la notion d'arborescence et les liens avec la structure générale de WINDOWS, tel le multi-fenêtrage, déjà cité.

Une particularité de l'affichage des menus, qui en principe devrait faciliter grandement le travail, a été souvent relevée comme mal comprise : il s'agit de l'utilisation de l'indication des commandes inactivées, apparaissant alors en gris. Outre le but premier de ce système, qui est d'éviter les fausses manœuvres - surtout si elles sont dangereuses - on peut lui attribuer une deuxième qualité, qui serait de faire acquérir à l'utilisateur des notions plus abstraites à partir de l'état licite/illicite d'une commande. Or les observations montrent plutôt que les éléments grisés suscitent l'étonnement, voire une incompréhension plus profonde, mise en évidence par des essais répétés (et infructueux) pour lancer la commande inactivée.

Nous verrons par la suite une manière d'aborder ce type de problème, au niveau formel (syntaxique), par une analyse en termes d'objets et de méthodes, dérivée de la théorie des Langages Orientés Objets, théorie qui a donné naissance à de nouveaux langages de programmation avec un certain succès (équipe de Rouen).

Un premier élément d'analyse : la structure fonction/argument

Plusieurs équipes, dont celle de Rouen⁶ appellent *fonction* la commande du logiciel et *arguments* les objets sur lesquels elle opère. Selon les logiciels, on rencontre deux séquences de travail voisines (les données étant l'ensemble des objets disponibles avant l'opération et le résultat, ce qui est produit par celle-ci) :

données - fonction - argument - résultats

données - argument - fonction - résultats

Une telle pluralité de séquences, quelquefois même au sein d'un seul logiciel, induit des confusions importantes chez les utilisateurs (observations Rouen et Toulouse).

Ce concept *fonction/argument* est central dans la maîtrise de l'utilisation d'un logiciel.

⁶ Composée de mathématiciens.

Il a été l'objet de deux sortes de réflexion chez les enseignants. L'équipe de Rouen propose d'aborder ce problème par une analyse basée sur les Langages Orientés Objet (voir chapitre 6). L'équipe de Toulouse étudie cette question en relation avec la conceptualisation de la notion de mémoire (voir chapitre 5).

LES FICHIERS, PROGRAMMES ET DONNÉES

Les fichiers sont des objets informatiques fondamentaux dont la manipulation est difficile à acquérir par tous les utilisateurs. En effet, ce mot *fichier* est l'un des termes dont le sens diffère peut-être le plus selon le contexte, informatique ou hors informatique. Des représentations préexistantes sont présentes chez nous tous, mais sont loin d'avoir des significations identiques : qui fait « spontanément » la différence entre une boîte contenant des fiches (communément appelée fichier), les fiches (individuelles), l'ensemble des fiches (physique) et l'ensemble des informations contenues dans cet ensemble de fiches (également appelé fichier !).

En conséquence, les tentatives (nombreuses) d'enseignement de ces notions par analogie avec les « fichiers matérialisés » sont susceptibles de semer une confusion peu propice à stabiliser des connaissances réellement opérationnelles sur les systèmes.

Une première approche peut faire apparaître trois niveaux de complexité : la nature des objets (programmes/données et par extension système/application/document) et leur désignation, les opérations sur ces objets, et leurs lieux de mémorisation.

Les observations de toutes les équipes montrent des difficultés et des erreurs dans ces trois registres.

Perception globale de la notion de fichier

☞⁷ Les résultats globalisés issus d'un vaste questionnaire (Toulouse, voir document n° 1 en annexes) montrent une perception approximative de ce qu'est un fichier de données (contenu/contenant/nature). Par contre, ce qui semble plus net se situe au niveau du fichier pris comme une entité (contenant, désignation, manipulations). ■

Cette dernière appréciation ne fait pas l'unanimité dans les autres observations.

7 D. Rosso.

Dénomination de l'objet fichier

Une des grandes difficultés se concrétise par la désignation complète du fichier, comprenant l'unité disque, les répertoires hiérarchisés (chemins), le nom du fichier proprement dit et son extension.

Cette entité s'est simplifiée avec l'évolution des systèmes d'exploitation, à la fois par le jeu des implicites (source de problèmes sous DOS), par les représentations graphiques des arborescences et par l'utilisation des menus.

Voici par exemple un relevé d'observations de l'équipe de Rouen (en 1991-92, sous DOS) :

☞ *Retrouver son fichier de données :*

- Une présentation a été faite rapidement de l'unité centrale et de ses périphériques. L'accès à un fichier exécutable ou de données a été abordé ;
- un mode d'emploi a été donné pour sauvegarder puis recharger un fichier personnel (avec désignation complète du fichier, unité de disquette comprise).

Exemples d'erreurs :

- oubli de la désignation de l'unité de disque où se trouve le fichier ;
- sauvegarde de fichiers sans préciser le nom, en écrasant le fichier de même nom : problème d'anticipation (confusion entre ENREGISTRER et ENREGISTRER SOUS).

Sauvegarde en fin de session

Elle est faite sans se préoccuper de l'unité cible. Il se trouve que c'était la bonne, par défaut, dans la session sur WORKS. Ce n'était pas le cas avec GÉOMÈTRE.

L'utilisation de WINDOWS simplifie et complique le problème des noms : il n'est plus nécessaire de désigner explicitement le fichier par tous ses paramètres grâce aux menus, ce qui fait disparaître la notion de chemin (d'un maniement si peu évident) ; inversement, les possibilités de mise en oeuvre dans une même application de documents de sources différentes repose le problème des structures de données. Nous pointons ainsi l'une des grandes questions conceptuelles, la notion **d'arborescence** et toutes ses instanciations (données, programmes...). ■

Les relations nom/contenu/icône d'un fichier

Nous avons remarqué que la représentation par icônes des objets dossiers et documents est d'une richesse qui dépasse largement les possibilités d'observation⁸ des débutants ; il ne suffit pas de différencier subtilement les graphismes pour faire comprendre aux utilisateurs les variations de caractéristiques des objets ainsi représentées et la signification des notions qui leur sont attachées.

Les opérations de réduction en icônes ou d'agrandissement ne sont pas davantage interprétées par les apprenants de manière satisfaisante, dans la mesure où ils ne peuvent pas *établir un lien de sens* entre les transformations purement graphiques et leurs conséquences sur le plan de la gestion des objets.

La nature des objets

Distinguer un fichier de données d'un fichier de programme pose problème pour des utilisateurs débutants, faute de pouvoir définir d'emblée ce qu'est un *exécutable* : les programmes proprement dits s'éloignent d'une observabilité directe, y compris dans les termes (programme/logiciel/application) ; les deux types de fichiers sont nécessaires pour démarrer (surtout s'il s'agit de la modification d'un document existant), mais il peut y avoir confusion sur les termes (traitement de texte/texte/document) ou sur les opérations de lancement (sous WINDOWS, par exemple, il suffit de cliquer sur le *document* pour lancer *l'application*).

Les supports et actions de lecture/écriture

Ces termes qui paraissent simples (simplement traduits de l'anglais...) sont un peu victimes des analogies trompeuses avec le support papier : « charger » a tendance à renvoyer à un déplacement de l'objet plutôt qu'à une recopie :

« si je l'ai chargé en mémoire, il n'est plus sur le disque »

avons-nous déjà entendu lors de nos observations sur le traitement de texte⁹. De la même manière, l'écriture ne fait pas prendre immédiatement conscience que le contenu précédent sera effacé. Il est vrai que les termes OUVRIR et ENREGISTRER sont plus proches de la réalité. Faut-il alors bannir du vocabulaire « lire » et « écrire » (en mémoire, sur disque ?).

⁸ Les divers travaux d'observation de situations d'apprentissage permettent de faire l'hypothèse que l'on ne peut saisir par l'observation que des éléments auxquels on s'attend « plus ou moins ».

⁹ Lire LEVY J.-F. op. cité.

Un enseignant¹⁰ consigne ainsi ses remarques sur cette même question, à la suite de l'observation d'un stage d'initiation au logiciel Hypertexte CONNEXIONS (voir chapitre 6) :

- ☞ *Vous venez de travailler avec le menu DÉCOUPAGE¹¹ sur le fichier MORCEAU.ASC. A votre avis, qu'est-il advenu à présent de ce fichier :*
- 1) *Il est intact sur le disque dur.*
 - 2) *Il est effacé du disque dur.*
 - 3) *Il est sur le disque dur mais modifié.*

Vous pouvez préciser les raisons de ce choix.

Cinq réponses (sur 12) sont correctes (le fichier est intact sur le disque dur). Trois personnes signalent même que c'est le fichier CFHYAP qui fait l'objet de modifications.

Aucune réponse n'envisage l'effacement du fichier.

Sept personnes pensent que le fichier est modifié avec des explications diverses qui vont toutes dans le même sens : à partir du moment où l'on manipule des informations, leur source est modifiée. Autrement dit, il ne semble pas y avoir de distinction entre les informations stockées sur le disque et leur copie temporaire en mémoire. La justification d'un stagiaire contredit peut-être cela : en effet, il explique la modification du fichier par une manoeuvre de sauvegarde, qui est modificatrice quand on travaille en réalité sur le même fichier. Ce qui n'est pas aperçu ici, c'est la capacité de la machine à lire et à écrire dans des fichiers différents au cours d'une manipulation perçue comme homogène par l'utilisateur.

Notre hypothèse supposant une conception fragile des opérations de lecture et d'écriture se vérifie ici assez largement :

Les hésitations des stagiaires, en particulier à propos du menu Découpage, montrent qu'ils ne sont pas très à l'aise avec le sens des opérations de lecture et d'écriture des fichiers sur disque.

« Lire » est facilement remplacé par « aller chercher ». Il s'ensuit que la lecture est perçue comme destructrice puisqu'une chose ne se trouve plus à l'endroit où l'on vient d'aller la chercher. Plus curieusement peut-être, « écrire » n'est pas facilement perçu comme destructeur. Il est vrai qu'on ne parle pas d'écrire mais de « sauver », ce qui est à l'opposé de la destruction dans la conception commune du monde. ■

10 C. EURIAT, Epinal-Nancy.

11 Le menu DÉCOUPAGE permet de lire des extraits sélectionnés de fichiers (sources) et de les copier pour constituer les données de l'hypertexte.

Ces questions renvoient à une problématique bien plus large que celle de l'entité fichier. Dès le début de l'étude, un collègue de Toulouse¹² avait cadré son travail à partir de l'interrogation globale suivante :

Où et comment existent les objets que je manipule ?

qui ouvre sur un problème « central » : La perception de la mémoire.

12 M. SAINT-MARTIN.

ANNEXES

DOCUMENT N° 1**QUESTIONNAIRE ÉLÈVES ET STAGIAIRES MAPPEN
(D. ROSSO, TOULOUSE)**

Ce document fait partie d'un questionnaire général comportant 15 parties relatives à tous les aspects du système micro-informatique utilisé par les élèves et les stagiaires.

Ces apprenants ont utilisé le nanoréseau et des compatibles PC, et ils ont écrit un récit sur plusieurs séances, avec sauvegarde, rappel de fichier, impression et perte de données par blocage des machines du nanoréseau.

Les questionnaires ont été remplis au cours des travaux. 155 questionnaires ont été rendus, 107 proviennent d'enseignants stagiaires, 49 d'élèves du premier cycle. 54 sont complets, 144 portent sur les notions de fichier, de données, de mémoire vive et de mémoire de masse. 17 sans réponse, 7 non compris, donc il reste 131 questionnaires exploitables.

Dans le tableau de la page suivante, tous les apprenants n'ont pas répondu, et ceux qui ont répondu n'ont pas forcément rempli toutes les réponses ; c'est pourquoi les totaux par lignes sont inégaux.

Commentaire : les difficultés naissent surtout :

- de la mauvaise différenciation du logiciel en général, pris au sens le plus large ;
- de la perception approximative de ce qu'est un fichier de données (contenu/contenant/nature).

Par contre ce qui semble plus net se situe au niveau du fichier pris comme une entité (contenant, désignation, manipulations).

CERNER LA NOTION DE FICHER

Un fichier de données peut être	toujours vrai	parfois vrai	faux	je ne sais pas
ce qui est visible à l'écran	19	45	9	6
créé	26	60	3	1
désigné par un nom	83		10	1
effacé sous DOS	63		25	7
le résultat d'un travail produit par un outil spécialisé	70		9	7
lisible sous un traitement de textes	13	40	3	3
lu sous DOS	19		39	9
mesuré. quantifié	30	26	8	7
modifié	61		26	2
perdu par coupure de courant	28		58	6
rangé dans un répertoire	70		15	4
récupéré sous DOS	31		38	24
renommé	79		6	6
tout ce qui est enregistré sur disque	45		31	11
un caractère	31		46	9
un dessin	20		38	15
un disque	59		11	10
un ensemble d'informations continues	54		18	16
un ensemble d'octets	6	35	16	25
un ensemble de codes	19		38	19
un nom	3		39	15
un nom désignant un ensemble de codes	45	38	4	4
un outil de travail	48		19	19
un texte	10		39	25