



HAL
open science

Logiciels pouvant impliquer des activités mathématiques à l'école élémentaire : typologie et enjeux didactiques.

Françoise Tréhard

► To cite this version:

Françoise Tréhard. Logiciels pouvant impliquer des activités mathématiques à l'école élémentaire : typologie et enjeux didactiques.. Histoire et perspectives sur les mathématiques [math.HO]. Université Paris VII, 1987. Français. NNT: . tel-01251146

HAL Id: tel-01251146

<https://theses.hal.science/tel-01251146>

Submitted on 5 Jan 2016

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

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

UNIVERSITE PARIS VII

DIPLOME DE DOCTORAT

SPECIALITE : DIDACTIQUE DES MATHÉMATIQUES ET DE L'INFORMATIQUE

THESE PRESENTEE PAR : Madame Françoise TREHARD

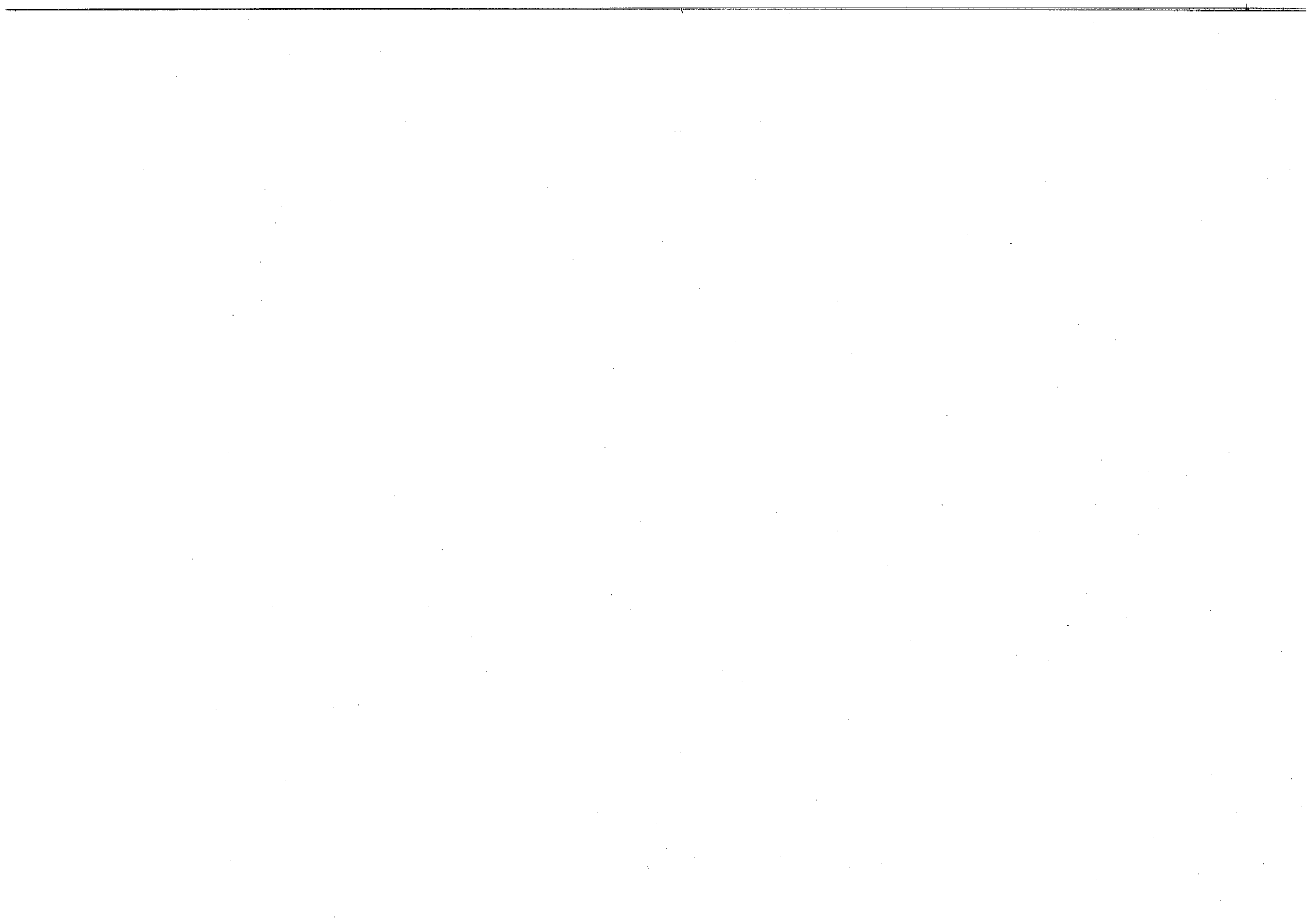
SUJET DE LA THESE Logiciels pouvant impliquer des activités mathématiques à l'école élémentaire : typologie et enjeux didactiques.

soutenue le 6 mai 1987

devant la commission d'examen

JURY :

Mr J. CHASTENET DE GERY	Président du jury
Mme A. ROBERT	Directeur de recherche
Mr J. ARSAC	Rapporteur
Mme J. ROGALSKI	Rapporteur
Mme H. BESTOUGEFF	Examineur
Mr R. GRAS	Examineur



Remerciements.

Je tiens tout d'abord à exprimer toute ma reconnaissance à Aline Robert pour la confiance qu'elle m'a témoignée, pour les encouragements et l'aide qu'elle m'a apportés ; sa rigueur m'a été précieuse.

Je remercie aussi chaleureusement Janine Rogalski et Régis Gras, pour l'attention qu'ils ont portée à ce travail, comme pour les nombreux conseils très appréciables qu'ils m'ont donnés.

Je remercie également Denis Butlen et Claire Lethielleux : les échanges que j'ai eus avec eux m'ont été très utiles.

De nombreux autres collègues ont rendu ce travail possible : je voudrais ici remercier toutes les personnes qui m'ont prêté des logiciels, - et tout spécialement André Deledicq -, ainsi que tous les enseignants qui m'ont permis de réaliser des observations.

Je remercie vivement Martine Lamy qui a bien voulu participer à la frappe de cette recherche sur une machine toute nouvelle pour elle, comme Odette Dieraert et Nadine Locufier qui en ont assuré le tirage.

Je remercie aussi Michèle Artigue, Directrice de l'I. R. E. M., pour les facilités qu'elle m'a apportées.

Enfin, je tiens également à remercier les professeurs qui ont accepté de faire partie du jury.

PRESENTATION

DE

LA

RECHERCHE



GENERALITES

1- SITUATION DE LA RECHERCHE

Cette recherche tente de constituer une synthèse d'expériences d'activités informatiques en liaison avec la didactique des mathématiques à l'école élémentaire.

La rédaction à proprement parler débute en 1983. Toutefois, ces expériences ont été menées à divers titres, depuis le début de l'année scolaire 1981-82.

Professeur à l'Ecole Normale d'Instituteurs de Paris, j'ai réalisé :

- l'encadrement de groupes d'instituteurs (en formation initiale et continue) menant une réflexion sur l'utilisation d'un micro-ordinateur :
 - . par l'utilisation de celui-ci avec les langages LOGO et BASIC,
 - . par la conception et l'expérimentation de logiciels,
 - . par l'analyse critique de divers logiciels,
 - . par l'observation d'enfants face à des produits informatiques.
- des observations d'adultes dans le contexte ci-dessus.
- des observations d'enfants de divers niveaux de l'école élémentaire utilisant des logiciels, le langage LOGO.

Animatrice à l'I.R.E.M Paris-Sud, dans le cadre de l'équipe informatique/école élémentaire, j'ai conçu et mené avec Denis Bullen l'expérimentation d'une situation utilisant un micro-ordinateur, portant sur la production et la maîtrise d'écritures multiplicatives au CE.

L'ensemble de la recherche s'inscrit dans le cadre d'une recherche globale de l'équipe informatique/école élémentaire de l'I.R.E.M Paris-Sud sur les apports spécifiques de l'utilisation d'un micro-ordinateur en mathématique.

2- MATERIEL DE LA RECHERCHE

Compte-tenu de ce qui précède, on peut dire que la recherche s'appuie

sur les éléments suivants :

- l'analyse de logiciels pour l'école élémentaire, sans expérimentation.
- l'aide à la conception de logiciels.
- l'analyse clinique et l'expérimentation de logiciels pour l'école élémentaire sur des cas ponctuels (enfants de l'école annexe à l'Ecole Normale, divers niveaux de classes).
- la conception et l'expérimentation avec D. Butlen de deux logiciels auprès d'une classe de CE 2 [classe de Mr Boyer, instituteur de l'école annexe ; 24 élèves ; expérimentateurs : Mr Butlen (PEN), Mme Tréhard (PEN), Mme Lépinay (IMF), Mr Boyer (IMF)].
- l'observation d'enfants travaillant avec le langage LOGO.
- l'observation d'enseignants de l'école élémentaire face aux problèmes de l'utilisation de langages, de logiciels, des finalités de l'emploi d'un micro-ordinateur.
- des lectures diverses.

3- VOCABULAIRE

3a Logiciel

Dans la recherche, le terme "logiciel" prend en compte et traite sans distinction, les langages et les autres produits informatiques.

Une étude sans différenciation initiale constitue un des principes de la recherche entreprise.

Ceci peut se justifier de deux façons :

- au niveau informatique, le terme "logiciel" désigne "l'ensemble des programmes, procédés et règles et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de l'information." (Terminologie de l'Informatique, arrêté du 22 décembre 81, Ministres de l'Industrie et de l'Education Nationale).

Le terme "logiciel" concerne donc aussi bien les logiciels d'exploitation que d'application. On peut d'ailleurs remarquer que certains produits désignés comme langages ont été réalisés à l'aide d'autres produits désignés comme langages : LOGO sous BASIC, LOGO sous LISP, etc ... Nous reviendrons ultérieurement sur les raisons qui peuvent conduire certains à particulariser le cas des langages, et sur des faits confirmant la validité de notre point de vue.

- au niveau de la position d'un utilisateur, on peut admettre que la place de ce dernier face à l'ordinateur peut-être considérée de prime abord comme indépendante du logiciel. Il ne voit et n'emploie en clair que les

périphériques de la configuration dont il dispose : pour lui, il s'agit toujours d'utiliser un périphérique d'entrée et de lire (au sens large) un périphérique de sortie. En ce sens, il n'y a pas de différence, qu'il soit utilisateur d'un logiciel d'exploitation ou d'application. Si des disparités interviennent, cela sera au niveau des diverses éventualités permises suivant les logiciels.

3b Langage

Le terme "langage" fera référence à langage informatique comme "ensemble des caractères, des symboles, des règles de syntaxe qui permettent de communiquer avec l'ordinateur" (dictionnaire Micro-informatique, Collection Micro-monde, Ed. Cedic Nathan).

Il ne sera néanmoins pas fait d'étude au niveau formel. Le point de vue utilisateur étant primordial dans notre recherche, il sera examiné de façon naïve les caractéristiques des logiciels attestés comme étant des langages, discernables du point de vue utilisateur.

4- CHOIX DES LOGICIELS INTERVENANT DANS LA RECHERCHE

Ce sont des logiciels au sens défini au paragraphe précédent.

La première partie de la recherche prend en compte plus spécialement un ensemble de neuf logiciels, parmi lesquels se trouvent :

- . des logiciels d'exploitation (LOGO) et d'application,
- . des logiciels portant sur le domaine numérique ou géométrique.

Ces logiciels sont présentés en annexe I.

Aucun logiciel n'est exclusivement ludique.

Aucun logiciel n'est du genre QCM, mais ce travail permettra néanmoins de rendre compte de leur existence.

La portée des résultats dépend de la représentativité des logiciels analysés.

Il aurait été possible de faire un autre choix, toutefois celui effectué a semblé proposer une diversité correcte. Depuis 1981, nous avons pu analyser selon la méthode qui va être développée de nombreux logiciels (surtout depuis 1985, année du plan "Informatique pour Tous") : ils ont confirmé les conclusions présentées à l'aide des neuf logiciels retenus en première partie. D'autres faits que nous évoquerons viennent également valider le point de vue de la recherche.

La deuxième partie de la recherche citera des logiciels variés, dont ceux intervenant en première partie.

OBJECTIFS DE LA RECHERCHE

1- DEFINITION DES OBJECTIFS DE LA RECHERCHE

L'objectif général de notre recherche est de contribuer à la mise en place d'une problématique de l'informatique-outil.

Dans ce but, nous nous fixons des objectifs intermédiaires :

- Etablir des grilles permettant l'analyse a priori de certaines spécificités de logiciels pouvant impliquer des concepts mathématiques objets d'enseignement à l'école élémentaire.

Les spécificités que nous envisagerons concerneront d'une part le contenu du logiciel et son organisation, d'autre part la communication que le logiciel propose à l'utilisateur avec le micro-ordinateur. Les critères retenus permettront l'analyse a priori, c'est-à-dire indépendamment d'un emploi en situation d'enseignement.

- Dégager une typologie à partir de ces grilles.

Cette typologie classera les logiciels indépendamment de leur utilisation effective pendant l'enseignement, ce qui explique le qualificatif "a priori" que nous lui appliquerons.

Pour préciser, l'analyse et la typologie a priori ne prendront pas en compte :

- les conditions d'utilisation du logiciel.
- la place, le rôle du logiciel dans l'apprentissage.
- l'adéquation des consignes émises, le cas échéant, par le logiciel ou son environnement.

On ne s'intéressera pas à la qualité technique du logiciel.

Soulignons que l'analyse et la typologie a priori ne prétendront pas être à elles seules un instrument de mesure de l'intérêt des logiciels.

- Confronter les différents types de logiciels dégagés et le vocabulaire informatique et didactique en usage.

- Emettre des hypothèses relatives aux enjeux de l'utilisation effective (pendant l'enseignement) de logiciels de chaque type, aux rôles impartis à la machine.

Il sera rendu compte de tests partiels.

- Amorcer une prospective de construction de logiciels.

Remarques :

- L'étude comparative des résultats fournis par les grilles d'analyse et des déclarations des auteurs des logiciels n'a pas été envisagée en raison du manque de documents à ce sujet, en 1983, au moment de la rédaction : les logiciels examinés étaient pour la plupart des prototypes sans documentation.

- La recherche entreprise concerne les logiciels de mathématiques pour le 1^{er} degré. Toutefois, son propos comme ses résultats semblent pouvoir se généraliser à d'autres cycles d'enseignement, et, pour nombre des questions abordées, à d'autres disciplines.

2- PERTINENCE DES OBJECTIFS RETENUS : A PROPOS DE LA DIVERSITE DES LOGICIELS

Depuis 1981, de nombreux micro-ordinateurs s'implantent dans les écoles. Plus encore, depuis 1985, année du plan "Informatique pour Tous", tous les établissements disposent de matériel et de logiciels.

Parmi les nombreux logiciels, spécialisés ou non pour l'enseignement, comment choisir pour la construction d'une séquence d'enseignement ?

En nous interrogeant sur l'approche usuelle de cette diversité, nous soulignerons que le matériel d'analyse des logiciels (1^{er} et 2^{es} degrés d'enseignement) est limité : il est essentiellement composé d'un ensemble de grilles, dites grilles d'analyse, et d'un vocabulaire. Les carences de ces référents en ce qui concerne une décision didactique, l'absence quasi-totale d'outils d'étude du rôle des logiciels dans l'enseignement des mathématiques nous paraissent justifier le propos de notre recherche.

2a Les référents terminologiques usuels

Nous commencerons par présenter la terminologie d'usage ; sa critique sera effectuée lors de l'examen des problèmes posés par l'ensemble des référents usuels, dans la suite du chapitre.

Nous nous limiterons au vocabulaire employé couramment en France, celui-ci incluant d'ailleurs des termes en provenance de l'étranger, ou y ayant des équivalents.

Nous ne ferons pas une étude statistique, mais un recensement argumenté par des textes qui, en France, constituent des bilans, et par

conséquent, reflètent un usage dominant ; certains d'entre eux ont un caractère institutionnel (Terminologie de l'Informatique, Rapport au Président de la République de J.C Simon, Rapport à la CEE de B. Schwartz, Catalogue des logiciels Informatique Pour Tous, ...).

Nous distinguons un vocabulaire d'appellation et un vocabulaire de classification.

⇔ LE VOCABULAIRE D'APPELLATION

Par vocabulaire d'appellation, nous entendons les dénominations destinées à différencier les produits informatiques.

L'usage d'un micro-ordinateur s'assortit de celui d'un "logiciel". Rappelons la définition de ce terme générique :

«Logiciel n.m. : Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de l'information.» (Commission de Terminologie de l'Informatique).

Des tentatives de distinction des divers objets de la classe "logiciel" ont eu lieu. De nouveaux termes sont apparus qui formulent des principes généraux, évoquent des domaines d'application. Nous mentionnerons essentiellement ceux qui sont susceptibles d'intéresser le secteur scolaire.

Citons tout d'abord le vocabulaire défini par la «Commission de Terminologie de l'Informatique». Son emploi est obligatoire pour les organismes publics français. Il s'agit de : «didacticiel», «progiciel», «tableur», «ludiciel» — et aussi de «langage formel», en référence aux divers logiciels traducteurs (LOGO, BASIC, ...).

«Didacticiel n.m. : Logiciel spécialisé pour l'enseignement.»
 «Progiciel n.m. : Ensemble complet et documenté de programmes conçu pour être fourni à plusieurs utilisateurs, en vue d'une même application ou d'une même fonction.»
 «Tableur n.m. : Programme de création et de manipulation interactives de tableaux numériques visualisés.»
 «Ludiciel n.m. : Logiciel de jeu.»
 «Langage formel n.m. : Langage qui utilise un ensemble de termes et de règles syntaxiques pour permettre de communiquer sans aucune ambiguïté (par opposition à langage naturel).»

Nous compléterons cette énumération par des termes que l'on rencontre dans des écrits de provenances diverses, ou qui apparaissent à

l'oral. Il est généralement malaisé d'en trouver une réelle définition. Nous nous contenterons ici de les situer tels qu'ils peuvent être appréhendés dans une première approche.

Les termes «logiciel-outil», «utilitaire», «boîte à outils logiciel» «outil logiciel», ne sont pas explicités. On constate qu'ils sont associés à certains progiciels, voire plus récemment aux langages, ou tout simplement à des ... logiciels.

Ainsi, le "Catalogue des logiciels Informatique pour Tous" (Novembre 85) entérine le terme «outil logiciels» tout en distinguant les «outils logiciels -traitement de textes, -gestion de données, -tableurs, -logiciels intégrés, -gestion de graphiques, -dessin assisté, -outils d'ÉAO, -gestion documentaire, -langages, -divers».

Nous remarquons que celui qui recourt à l'une de ces désignations entend signifier une fonction d'aide à un utilisateur qui souhaiterait régler un problème en en conservant une certaine maîtrise. Si ce qualificatif peut se concevoir sans trop d'analyse pour certains produits comme les traitements de texte, ou les langages¹, il est malaisé aux enseignants d'expliquer son emploi pour des logiciels à vocation plus spécifiquement didactique. Dès lors, ils insistent sur le mot "outil".

Un «langage auteur» est un progiciel pour la réalisation de didacticiels.

Signalons que la rubrique «outils logiciels-outils d'ÉAO» évoquée ci-dessus comprend les «langages auteur».

Un «logiciel de simulation» veut permettre ce que son nom indique, ... mais, de quoi s'agit-il ? Il est très difficile de repérer une véritable communauté de sens pour un terme employé dans des contextes très variés.

1. En 1982, au début de cette rédaction, «outil logiciel» n'aurait pas été associé à «langage». «Langage» était réservé aux logiciels traducteurs des "reconnus" langages (Basic, LOGO, ...) et constituait un titre de noblesse. Pour les non-informaticiens, un langage n'était pas un logiciel ! Peu auraient osé assimiler un tableur et le langage X. Ce n'est que lentement, et avec le développement et le succès de certains logiciels, que les choses ont évoluées. Notre parti pris de départ - analyse des logiciels à l'aide de la même grille, qu'ils soient d'exploitation ou d'application - se trouve maintenant fréquemment confirmé, ici au niveau de la nomenclature proposée par le catalogue Informatique Pour Tous.

La publication "Imagiciels enseignement des mathématiques illustré par ordinateur" (INRP) donne son point de vue sur le terme «imagiciel» :

"Il existe peut-être quelque part une définition précise du terme «imagiciel», mais nous prendrons la liberté de dire ici comment nous l'entendons: il s'agit d'une illustration par ordinateur, à base de graphique, dans un but d'enseignement. Ce n'est donc pas autre chose que la prolongation, par des moyens informatiques, des aides pédagogiques plus ou moins classiques à base d'images: figures sur papier ou tableau noir, diapositives, films... L'ordinateur ne supplante pas actuellement ces autres moyens, car leurs performances, dans le cadre de leurs spécificités, sont assez bonnes. Il empiète cependant sur leur domaine et surtout fournit un nouveau mode d'illustrations ayant de grandes possibilités dues à ses particularités." (p.16-17)

Le titre du chapitre V du livre de S. Papert "Jaillissement de l'esprit Ordinateurs et apprentissage" est "Micromondes: incubateurs de savoir". Il est à notre connaissance impossible de trouver une définition précise du terme «micromonde», propre au domaine de l'informatique éducative. Le mot apparaît soudain dans la littérature et est susceptible de désigner un environnement matériel, logiciel, intellectuel. Ce dernier autoriserait un individu à exprimer sa propre représentation d'un problème, et y réagirait. Ainsi, un micromonde se découvrirait, se conquerrait.

- Dans "Recueil de pratiques pédagogiques autour de LOGO" (C. Berdonneau), on trouve "micro-monde", inscrit en italique à la page 3, parmi des phrases explicatives du type de celles que nous avons proposées :
"Cela conduit à fournir aux enfants des outils très simples contrôlés par ordinateur, choisis de manière à permettre la découverte d'un *micro-monde* : dans ce milieu où il peut librement conduire ses propres expériences pour en arriver à le comprendre et à le maîtriser, l'étudiant est amené à élaborer et à essayer des modèles de pensée : ainsi, non seulement l'élève construit lui-même sa propre connaissance, mais en même temps il est son propre agent de développement."

- Il en est de même chez d'autres auteurs, dont S. Papert où le mot apparaît dans son ouvrage de référence, "Jaillissement de l'esprit : ordinateurs et apprentissage", pour la première fois entre guillemets en page 24.

Jusqu'à présent, «micromonde» a été employé essentiellement par des partisans de LOGO, en référence au logiciel LOGO ou à des logiciels écrits sous LOGO. Toutefois, on peut noter parmi les développements actuels de l'intelligence artificielle la création de jeux d'aventure (en particulier, sous le logiciel PROLOG), où l'utilisateur crée son «monde» à l'aide de «boîtes à outils».

Signalons que si le terme «micromonde» désigne un contexte d'apprentissage, il devrait pouvoir concerner, le cas échéant, d'autres logiciels que LOGO, ou plus récemment PROLOG, comme des environnements non-informatiques (ce que n'exclut pas S. Papert).

Enfin, nous n'oublions pas que ce terme est issu de la recherche en intelligence artificielle. Selon H. Dreyfus (Intelligence artificielle: mythes et limites), il a été créé pour désigner une méthode consistant à travailler sur un univers clos et restreint afin de circonscrire un domaine (par exemple celui du dialogue), indépendamment de considérations didactiques ou psychologiques.

"Les chercheurs de l'I.A n'ont certes pas tenté de camoufler le fait que le seul univers clos et restreint de SHRDLU permet cet apparent début de compréhension. Ils ont même créé un terme pour désigner la méthode utilisée ici par Winograd afin de circonscrire le domaine du dialogue. Ils appellent cela travailler sur un «micro-monde». Et, dans un texte à usage interne, au M.I.T, Minsky et Papert reconnaissent ouvertement en 1970 : Chaque modèle considéré – nous dirons plutôt «micromonde» – est très schématique en soi. Il s'agit en quelque sorte d'un univers imaginaire, en réduction, et dans lequel les choses sont tellement simplifiées que l'on ne peut pratiquement rien dire d'elles qui ne serait faux ou inexact une fois transposé dans le monde réel.

Mais ils s'empressent d'ajouter :

Cependant, ces micromondes nous paraissent présenter tant d'intérêt que nous allons consacrer une large part de nos efforts à en réunir une collection et rechercher comment exploiter les notions et les idées que peuvent suggérer ces modèles, tout en éliminant les inconvénients dus au fait qu'ils ne sont pas conformes à la réalité"

("Intelligence artificielle, mythes et limites", H L. Dreyfus, p.120-121)

"Dans diverses publications antérieures, Minsky et ses collaborateurs prenaient bien soin de se démarquer de certains chercheurs en simulation cognitive – tels Simon, par exemple – qui présentaient leurs programmes comme des travaux de psychologie. Ils insistaient bien, pour leur part, sur le fait que les projets conçus au M.I.T se donnaient pour but de «réaliser des machines intelligentes, sans se préoccuper de savoir s'il fallait ou non les concevoir à l'image de l'esprit humain.» Or, voici qu'en 1973, dans un ouvrage intitulé *Artificial Intelligence* et dressant le bilan des travaux effectués au M.I.T entre 1967 et 1972, Minsky et Papert présentent au contraire les résultats obtenus comme une contribution à la psychologie."

("Intelligence artificielle, mythes et limites", H L. Dreyfus, p.132-133)

⇔ LE VOCABULAIRE DE CLASSIFICATION

Nous nous intéressons ici aux termes qui nous semblent le plus souvent utilisés pour qualifier les relations enseignement/ordinateur.

Nous parlons de vocabulaire de classification, car, comme nous le verrons, il fonctionne pour de nombreux enseignants comme moyen de classement des logiciels par confrontation avec les principales théories de l'apprentissage.

Nous allons en effectuer une présentation susceptible de faire apparaître ce qui nous semble caractériser ce moyen de classement : non fondé sur des critères rationnels, il est en outre de type bipolaire ; la suite du chapitre montrera qu'il risque donc de conduire à rejeter sans analyse effective toute une gamme de produits informatiques.

Le terme «EAO» (Enseignement Assisté par Ordinateur) est ainsi défini, à partir des travaux de l'AF CET, dans le rapport "L'éducation et l'informatisation de la société" (J.C. Simon) :

"Le terme (abrégé) d'EAO prévaut chaque fois que l'enseigné ou l'enseignant est en contact avec l'ordinateur. L'AF CET, société française des ingénieurs et scientifiques informaticiens, a précisé la terminologie, grâce à son groupe de travail «Aide de l'informatique à l'enseignement» :

- « *L'EAO au sens large* désigne toutes les applications utilisant l'ordinateur à des fins didactiques, qu'elles soient conversationnelles par l'intermédiaire d'un terminal ou réalisées en temps différé» :
 - les systèmes de dialogues interactifs ;
 - les banques de données ;
 - les procédés de docimologie (évaluation des élèves) ;
 - la gestion, soit les problèmes de gestion proprement dits (finance, personnel, horaires, etc.), soit la progression du niveau des enseignés et leur orientation scolaire et professionnelle (en anglais, Computer Managed Instruction, CMI) ;
 - le recherche pédagogique, utilisant l'informatique.

- « *L'EAO au sens strict* comprend uniquement les utilisations conversationnelles dans lesquelles l'étudiant est en communication directe avec l'ordinateur par l'intermédiaire d'un terminal.»

(En anglais appelé Computer Assisted Instruction, CAI). Il est de deux types :

- *directif*, l'enseigné est guidé par le système suivant ses réponses, en général du type QCM (question à choix multiple) ;
- *non directif*, on cherche à ce que l'enseigné soit le plus libre possible."

("L'éducation et l'informatisation de la société. Rapport à Monsieur le Président de la République.", J.C. Simon, p. 202).

Selon ce texte, le terme EAO devrait prévaloir dès qu'il s'agit d'une utilisation de l'ordinateur à des fins d'enseignement.

On note aussi que le rapport Simon distingue plusieurs sortes d'EAO, dont en particulier :

- l'EAO au sens strict de type directif, ce que certains (par exemple, B. Schwartz dans "L'informatique et l'éducation", p.27) appellent «EPO» (Enseignement Programmé par Ordinateur).
- l'EAO au sens strict de type non directif, auquel participe, selon le rapport Simon, l'«apprentissage autonome», identifié au «projet LOGO» (p. 206-207).

Ce vocabulaire, proposé par le rapport Simon pour refléter des modalités d'intervention de l'ordinateur dans l'enseignement, ne semble pas faire l'unanimité. Des controverses existent et peu d'enseignants acceptent de reconnaître en LOGO une forme d'EAO. Ainsi, dans un souci de particularisation du cas LOGO, C. Berdonneau écrit, en contradiction avec le rapport Simon, dans "Recueil de pratiques pédagogiques autour de LOGO" :

"Développé depuis maintenant plus de dix ans, originellement aux Etats-Unis, et ensuite diffusé dans bien d'autres pays, le projet LOGO se distingue de la plupart des utilisations d'un ordinateur dans l'enseignement : il ne s'agit pas d'enseignement géré par ordinateur (*computer managed instruction - CMI -*), qui, comme son nom l'indique, se propose d'aider l'enseignant ou l'administration scolaire dans la gestion de l'apprentissage (évaluation, diagnostic, aiguillage, tenue des dossiers scolaires, organisation du temps scolaire ...) ; il ne s'agit pas non plus au sens strict du terme d'enseignement assisté par ordinateur (*computer aided instruction - CAI -*), où la quasi-totalité du "savoir" est déjà collectée et organisée par un tiers (le rédacteur du didacticiel) et présentée à l'élève dans le but de lui faire acquérir des connaissances et/ou des comportements.

Il s'agit au contraire d'utiliser l'ordinateur comme un instrument pédagogique, pour favoriser une formation générale de l'élève : les dénominations sont variables : *enseignement assisté intelligent, apprentissage autonome, pratique active* ("Recueil de pratiques pédagogiques autour de LOGO", C. Berdonneau, p.1).

Signalons que, pour d'autres, le terme «EIAO» (Enseignement Intelligent Assisté par Ordinateur) s'applique à l'usage d'un système expert d'enseignement, avec des variantes : l'enseignement peut être confié complètement au système expert qui intègre notamment des heuristiques ; ou encore, le système expert, pour réaliser une meilleure interface utilisateur, laisse parfois la "main" à celui-ci.

Comment expliquer ces divergences, alors qu'on peut estimer parfaitement raisonnable d'appeler "Enseignement Assisté par Ordinateur" "toutes les applications utilisant l'ordinateur à des fins didactiques" ?

Deux théories de l'apprentissage opposées soutiennent respectivement l'EPO (EAO directif) et le projet LOGO (forme d'EAO non directif). Succinctement :

- L'EPO est basé sur le courant psychologique behavioriste et propose des schémas stimulus-réponse indépendants de la personne qui visent à renforcer des comportements. L'évaluation est de type normative.

Nous ferons appel au rapport Schwartz ("L'informatique et l'éducation") pour présenter les principes de l'EPO :

"L'enseignement programmé est fondé sur les principes suivants :

- Premier principe : l'étudiant apprend avec d'autant plus d'efficacité qu'il avance en passant par une série de petites étapes soigneusement préparées. Si chaque étape à franchir est courte, les erreurs sont moins probables (principe des petites étapes).
- Second principe : les résultats de l'étude sont d'autant meilleurs qu'il y a participation active de l'étudiant à son travail. Résoudre soi-même un problème d'arithmétique vaut mieux que toute lecture ou écoute d'explications (principe de la participation).
- Troisième principe : les résultats de l'étude sont d'autant meilleurs que l'étudiant peut vérifier sa réponse immédiatement (principe de renforcement).
- Quatrième principe : chaque étudiant peut franchir une étape aussi vite ou aussi lentement qu'il choisit de le faire (principe de l'allure personnelle).

L'exemple simplifié ci-dessous illustre les différents principes ; il est relatif à la «commutativité de l'addition».

Les quelques items et/ou phrases seraient les suivants :

1. $2+4=6$ $4+2=6$
2. $9=4+5$ $9=5+4$
3. L'ordre dans lequel se fait l'addition ne joue aucun rôle. C'est la règle de commutativité.
4. $a+b=c$ $b+a=?$
5. Le fait que $a+b=b+a$ se nomme ? ».

Un dernier principe de l'enseignement programmé est qu'il soit validé : un enseignement programmé est dit validé pour une population quand/si 90% des élèves comprennent seuls et bien 90% des items. Cette règle suppose deux conditions : que 60% au moins des questions posées ne soient pas nouvelles pour l'élève, afin que celui-ci ne soit pas perdu par la trop grande quantité d'inconnues, et que «toutes» les mauvaises réponses aient été étudiées et leur «réponse» prévue. C'est donc un enseignement

de «succès».

Dans le cas de l'enseignement programmé sur ordinateur, le contenu «programmé» est mis sur l'ordinateur. La ou les réponses de l'ordinateur peuvent être données par écrit, ou oralement, ou par dessin. Dans tous les cas, il y a dialogue, et si, comme on l'a dit, le dialogue est soigneusement préparé, l'élève peut ainsi apprendre et «pratiquer» (*drill and practice*) indéfiniment, par lui-même, à son propre rythme, et sans l'appui de personne."

("L'informatique et l'éducation", B. Schwartz, p. 27-29).

Au départ, l'EPO était linéaire (Skinner), puis sont apparus des programmes ramifiés (Crowder).

Une connexion s'est établie entre le terme «didacticiel» de la nomenclature d'appellation et l'EPO, ceci vraisemblablement en raison des ordres d'apparition respectifs des éléments de vocabulaire et des types de produits informatiques utilisables dans le cadre de l'enseignement. En ce sens, on peut dire que l'EPO se matérialise par des didacticiels.

• **Le projet LOGO** se réclame d'une psychologie constructiviste. Sur la base des thèses de Piaget, il revendique pour l'apprenant la construction de son propre savoir par l'action, la réorganisation, l'évaluation formative (analyse des erreurs, des démarches, des difficultés).

Nous évoquerons les principaux traits du projet LOGO, tels qu'ils nous apparaissent à la lecture de divers textes, dont celui de S. Papert :

- l'apprentissage se veut sans instruction, au sens de "sans consigne". Ceci se traduit par le droit et la nécessité pour l'enfant de travailler à son propre projet.

- il convient de définir un environnement (ordinateur, langage LOGO, périphériques) susceptible de proposer des "objets pour penser avec", permettant l'exploration de micromondes, sans apprentissage spécifique du langage de programmation.

- l'apprenant doit avoir la possibilité de mener ses expériences, de procéder par essai-erreur, de réaliser un apprentissage autonome, d'être épistémologue, l'adulte ayant un rôle de "facilitateur".

Aujourd'hui, le projet LOGO se matérialise par un langage, LOGO, un ensemble de périphériques divers, la mise en place des «micromondes». On peut ainsi mentionner de façon non exhaustive : le micromonde de la géométrie de tortue, des micromondes dans les domaines de la musique, du texte, de la physique, ... ("Recueil de pratiques pédagogiques autour de LOGO", C. Berdonneau ; "Le jeune enfant et le micro-ordinateur", Chauvin-Eimerl, ...).

Dès lors, on comprend mieux les querelles de vocabulaire. Il y a assimilation de l'EAO à l'EPO, — l'EPO constituant la première, et unique forme d'EAO pendant quelques années¹—, et souci de démarcation de LOGO de l'EAO.

Le débat se résume souvent par le versus : EAO, enfant programmé/ LOGO, enfant programmeur. Citons à ce sujet S. Papert :

"Dans bien des écoles, aujourd'hui, l'expression «enseignement assisté par ordinateur» signifie que l'ordinateur est programmer pour enseigner à l'enfant. On pourrait dire que l'ordinateur sert à programmer l'enfant. Dans ma vision des choses, l'enfant programme l'ordinateur et, ce faisant acquiert la maîtrise de l'un des éléments de la technologie la plus moderne et la plus puissante, tout en établissant un contact intime avec certaines des notions les plus profondes de la science, des mathématiques, et de l'art de bâtir des modèles intellectuels."
("Jaillissement de l'esprit : ordinateurs et apprentissage", S. Papert, p. 15-16).

Enfin, nous soulignerons que le projet LOGO existe en tant que mouvement social (travaux de recherche, conférences, colloques, associations) : ceci atteste l'ampleur d'un phénomène qui dépasse les seules raisons logiques. Or, nous remarquons que si des interrogations de plus en plus nombreuses se font jour à propos de LOGO et de sa pertinence, une réactualisation de sa problématique interne et du caractère social qui l'accompagne semble s'effectuer autour du logiciel PROLOG.

2b Approche usuelle de la diversité des logiciels

Nous tenterons tout d'abord une critique de l'ensemble des moyens offerts à l'enseignant pour effectuer un choix de logiciels, lorsqu'il souhaite introduire l'ordinateur dans le cadre d'un enseignement. Ceci nous permettra ensuite d'évoquer comme conséquence d'une approche usuelle de la diversité des logiciels basée sur des moyens non pertinents, le risque d'une appréciation superficielle du rôle des moyens informatiques dans l'enseignement.

¹ Deux dates pour l'EAO en France :

- 1965 : EPO avec des didacticiels artisanaux (rapport Simon).

- 1976 : premier LOGO disponible à l'INRP ("Recueil de pratiques pédagogiques autour de LOGO", C. Berdonneau).

⇒ LES MOYENS D'IDENTIFICATION DES LOGICIELS

Confronté au foisonnement actuel de logiciels, l'enseignant dispose de moyens d'identification de leur diversité présentant des carences eu égard à une décision didactique.

Ainsi, il existe deux types de référents en matière de logiciels :

- un vocabulaire d'appellation et un vocabulaire de classification.
- un ensemble de grilles dites d'analyse.

Le vocabulaire, que nous avons présenté précédemment, appelle des remarques.

Le vocabulaire d'appellation (termes en ...ciel, langage, micromonde, logiciel-outil, ...) formule des principes généraux. N'étant pas défini en termes fonctionnels didactiques, il ne permet pas, en particulier, de préciser les tâches dévolues à l'enseigné ou à l'enseignant. De plus, comme il ne correspond pas non plus à l'énonciation de spécificités intrinsèques des logiciels, son emploi raisonné s'avère improbable.

Le vocabulaire de classification (EAO, EIAO, EPO, projet LOGO...) pose plusieurs problèmes.

La terminologie d'ensemble EAO du rapport Simon, reprise par le rapport Schwartz, nous paraît acceptable, ne serait-ce que pour le sens de EAO. Nous émettrons néanmoins des réserves :

- dans ces rapports, la distinction entre EAO directif et non directif est arrêtée au niveau d'idées générales. Les modalités de chaque type d'EAO sont explicitées en recours au vocabulaire d'appellation (simulation, micromondes, ...) dont nous avons souligné les lacunes.

- ce vocabulaire n'est donc pas un outil d'étude des logiciels ; comme nous l'avons déjà remarqué, pour de nombreux enseignants, il ne reflète pas non plus de façon satisfaisante leur prise en compte de l'informatique. Dès lors, dans les faits, le désaccord terminologique nous paraît déboucher sur des renforcements de sens du lexique primitif (voire son enrichissement dans un contexte analogue - cf EIAO -) : ce dernier semble devenir une expression des principales théories de l'apprentissage dans un nouveau domaine, implicitement conçu comme autonome et auto-suffisant, celui de l'ordinateur comme moyen d'enseignement. En symbiose avec le vocabulaire d'appellation, il consacre plus la négation du rattachement d'un logiciel à telle théorie de l'apprentissage que sa filiation fondée à telle autre (EIAO, négation de EAO).

Le lecteur pourra prendre connaissance en annexe II de quelques grilles d'analyse parmi celles d'origines diverses que nous avons consultées : grilles d'examen de la pertinence pédagogique d'un logiciel mises au point par

des enseignants, des associations d'enseignants (EPI par exemple), l'institution (grille de l'ADI ou du plan IPT) ; grilles susceptibles de renseigner sur la fiabilité commerciale d'un logiciel d'enseignement (grille du CESTA par exemple).

Dans le cadre de ce travail, nous ne ferons pas d'exposé approfondi de leur examen ; nous nous limiterons à une analyse macroscopique sans démonstration, telle qu'elle peut être faite en première lecture.

Nous avons remarqué que les grilles d'analyse interrogent essentiellement sur des aspects techniques, des recours du logiciel à des spécificités du micro-ordinateur, des conditions d'emploi, voire des qualités ou options pédagogiques.

Nous dirons qu'elles font fréquemment intervenir des problèmes qui ne sont pas pertinents, ou qu'elles n'offrent pas de moyens de justification rationnelle des réponses apportées aux questions qu'elles posent : ainsi, tout logiciel fait appel à des spécificités du micro-ordinateur ; ou encore, la question "L'élève est-il actif ?" n'offre pas d'intérêt véritable, parce que les réponses qu'on peut apporter ne sont pas justifiables, ne serait-ce qu'à cause de l'imprécision de la question.

Nous les qualifierons plus de supports à l'émission de jugements de valeur (grilles d'évaluation sans fondement) que d'instruments d'analyse (grilles de connaissance raisonnée pour une appréciation des tâches).

En outre, quoique n'explicitant en rien leurs structures, elles ne concernent que certains logiciels, notamment ceux qualifiés de didacticiels¹. Dès lors, qu'en est-il des autres logiciels ?

En conclusion, l'enseignant ne peut recourir à des outils opératoires, indispensables à une étude rationnelle des logiciels dans la perspective de choix didactiques. Cette situation n'est pas sans effet sur sa conduite.

⇒ UN PROTOCOLE DE CHOIX DE LOGICIELS LIE AUX OUTILS DISPONIBLES

Nous exposerons un schéma de fonctionnement qu'il est possible de rencontrer, sans prétendre qu'il s'applique systématiquement, ceci dans le but de mieux cerner l'intérêt d'une autre démarche.

Un tel schéma précisera sur quoi peut apparaître basée une conduite de choix : des observations variées montrent qu'il existe des conditions objectives, accentuées du fait des moyens d'identification disponibles, qui font que les déterminants inconscients, idéologiques sont très forts, et interviennent contre des analyses rationnelles.

1. En ce qui concerne les «didacticiels» d'EPO, il existe des schémas descriptifs (schémas crowdériens).

Approfondir les réflexions qui suivent serait une nécessité ; toutefois, ceci sort du cadre de notre propos qui porte sur les logiciels, et non sur une étude de cas d'enseignants.

La nécessité où se trouve un enseignant de choisir, mais aussi de communiquer, peut l'engager fortement à exploiter les moyens d'identification existants. Il se sent proche d'une théorie de l'apprentissage, ses décisions interviennent en conséquence. Sur cette base cohérente, il lui reste à déterminer les produits conformes.

Dès lors, l'agrément d'un logiciel peut être assujéti à sa répertoriati on selon le double lexique.

On constate que l'alternative théorie piagétienne VerSus théorie behavioriste trouve des équivalents dans les deux nomenclatures. Comme nous l'avons déjà remarqué, elle s'assimile à LOGO, enfant programmeur VS EAO, enfant programmé.

En outre, les deux champs terminologiques se sont construits avec les premières tentatives en matière d'informatique pour l'enseignement. De ce fait et en fonction de caractéristiques de produits datés, leur liaison, sans argumentation, s'actualise en langage-micromondes VS didacticiels : le genre «didacticiel» regroupe alors tout ce qui n'est pas «langage-micromonde», une relative exception étant faite pour «logiciel-outil».

Ainsi, un vocabulaire, dont on connaît les lacunes, tient lieu de critère de ce que l'on ne peut appeler une analyse : sera retenu ce qui, au plan terminologique, correspond au parti de l'enseignant dans l'opposition relative aux théories de l'apprentissage.

Dans ce contexte, comment l'enseignant procède-t-il pour attribuer une dénomination à un logiciel ?

La dénomination peut provenir d'autres que lui (origine commerciale, documentation, rapport oral de collègues), l'enseignant la reprenant à son compte.

Dans le cas contraire, le parcours peut être le suivant : il y a tout d'abord prise d'informations par visualisation de quelques produits. Cette phase fige le lexique à partir de traits superficiels, s'ils ne sont pas caricaturaux. Intervient fréquemment la perception de signes significatifs de logiciels pionniers, dont le vocabulaire de classification et d'appellation est arrêté. Ensuite, la reconnaissance des indices dégagés auparavant (conjuguée au respect de langage-micromonde VS didacticiel) assurera l'attribution à tel logiciel de tel terme de la nomenclature d'appellation, comme son affiliation selon le vocabulaire de classification.

Avec une telle démarche, la dénomination d'un logiciel se fait sans

fondement rationnel. Elle relève de l'appréciation individuelle, la décision de choix lui étant par ailleurs subordonnée. On rencontre ainsi des inconditionnels d'un genre exclusif de logiciels («didacticiels» ou exclusif «LOGO»), retenant ledit genre comme seul susceptible de présenter un intérêt pour l'enseignement.

Nous proposerons en hypothèse une explication complémentaire concernant ce procédé de choix.

Nous ne nions pas qu'il existe des logiciels, qui, employés pour un apprentissage, sont caractéristiques d'une méthode "j'apprends, j'applique". Ils relèvent, dans ce type d'usage, de méthodes pédagogiques que nous ne retenons pas. Nous ferons l'hypothèse que, s'ils sont destinés à l'apprentissage, l'idéologie que l'on peut percevoir à leur propos est celle d'une vision de "l'enfant qui apprend et applique", de "l'enfant qui répète, qui imite".

Mais, nous ferons aussi l'hypothèse que le projet LOGO, n'est pas exempt de toute idéologie¹. Il s'agirait d'une idéologie prônant "l'enfant créateur"². Ceci peut déjà transparaître dans les citations que nous avons faites ; la suite de notre travail en montrera d'autres marques.

En ce sens, l'opposition fondée sur une vision psychologique différente de l'enfant en recouvrirait une autre d'ordre idéologique, enfant créateur VS enfant enseigné. S'affronteraient une idéologie permissive, libertaire et une idéologie répressive.

2c Conclusion

Il existe des outils externes applicables à l'examen des logiciels : des grilles d'analyse qui sont plutôt des moyens d'évaluation sans fondement, et un vocabulaire d'usage.

Les produits originaux ne résistent pas à une analyse effectuée avec les seuls moyens évoqués. Il y a difficulté à imaginer qu'il puisse exister d'autres logiciels que ceux entrant dans le cadre de l'alternative EPO VS LOGO. Les tentatives de singularisation par création de nouveaux termes (imagiciels, logiciel-outil, ...) sont relativement vouées à l'échec, car elles ne mettent pas en place des critères de distinction probants.

1. Dans certains écrits, discours, ou dans certaines pratiques. Nous nous y référerons pour étayer notre hypothèse. Nous recourerons aussi à d'autres textes, ceux d'A. Rouchier par exemple, qui dénotent une position différente, justifiant en quelque sorte ce que nous avançons.

2. On peut se demander si les événements de Mai 1968 n'ont pas occasionné ou entériné une rupture concernant cette vision que peut avoir l'adulte, ici enseignant, de l'enfant.

Projeter une vision irrationnelle, datée, voire idéologique sur une situation neuve fait obstacle à une juste appréciation du rôle de l'ordinateur dans l'apprentissage. En outre, une telle méthode ne nous paraît pas s'intéresser aux enjeux et conditions d'usage des logiciels.

C'est pourquoi, la situation de contrôle du phénomène informatique est à modifier.

Il nous paraît important de déterminer des outils de connaissance des logiciels qui, permettant de dépasser le simple jugement par assimilation, soient des aides à la décision didactique. Ils seraient décisifs pour la perception du rôle des logiciels dans l'enseignement. Ils pourraient statuer sur le vocabulaire actuel, comme sur la création conforme de nouveaux termes. Ils pourraient également permettre de banaliser le débat sur les logiciels.

Notre recherche espère engager une réponse, ne serait-ce que partielle, à ces problèmes.

HYPOTHESES ET METHODOLOGIE

1- HYPOTHESES

Nous retenons les hypothèses suivantes :

- *La communication avec le micro-ordinateur est un élément spécifique : peut-être serait-il possible de l'envisager en tant que variable didactique.*

- *Il y a dépendance entre le type de contenu et le type de communication proposés par un logiciel.*

En conséquence, pour rendre compte d'un logiciel, il convient de s'intéresser :
 . à son contenu,
 . à la communication qu'il propose à l'utilisateur.

2- METHODOLOGIE

2a Préalable

Pour répondre à la problématique de choix rationnel de logiciels dans un but didactique, nous séparons l'objet (le logiciel) de son usage.

Autrement dit, nous distinguons l'analyse d'un logiciel de son évaluation, et retenons le principe d'une évaluation en trois étapes :

- analyse a priori du logiciel, indépendamment de son utilisation effective pendant l'enseignement, sans émission d'hypothèses relatives aux caractéristiques et enjeux de situations qui l'emploieraient.

- utilisation des résultats de l'analyse a priori pour cerner tâches et enjeux didactiques.

- confrontation des résultats de l'analyse a priori aux acquis didactiques sur le sujet mathématique en question ; rejet ou adoption du logiciel pour la mise au point de séquences ; expérimentation.

Nous allons préciser notre méthodologie, en tenant compte de cette position et de nos hypothèses.

2b Organisation de la recherche

En ce qui concerne, l'organisation générale de la recherche, ceci signifie que nous présenterons :

- dans une première partie (Partie I), l'élaboration d'une typologie a priori.
- dans une deuxième partie (Partie II), ce que nous pensons pouvoir dire des enjeux didactiques de l'emploi de logiciels au cours de séquences d'enseignement.

La partie I sera l'occasion de poser des questions, auxquelles la partie II apportera des éléments de réponse.

2c Méthodologie spécifique à l'élaboration d'une typologie a priori (Partie I)

Une analyse a priori autorisant l'accomplissement des deux dernières étapes de l'évaluation nous paraît être la pierre angulaire de l'étude, et a posteriori, de la construction de logiciels utilisables dans l'enseignement.

Dans la première partie (Partie I), pour l'élaboration de la typologie a priori, nous définirons des grilles d'analyse que nous espérons originales : en relation avec notre cadre théorique, la didactique des mathématiques, elles ne feront appel qu'à des critères rationnels directement liés à la didactique des mathématiques.

Compte-tenu des hypothèses que nous avons énoncées en début de chapitre, nous proposerons :

- une grille, que nous désignons par CONT, pour l'analyse a priori des logiciels du point de vue du contenu, indépendamment de la communication.
- une grille, que nous désignons par COM, pour l'analyse a priori des logiciels du point de vue de la communication proposée à l'utilisateur, indépendamment du contenu.

La méthodologie concernant l'établissement à proprement parler de la typologie a priori repose, elle aussi, sur nos hypothèses et se définit comme suit :

1 Etablissement d'une typologie du point de vue du contenu : typologie CONT

- Construction d'une grille d'analyse a priori du contenu du logiciel, indépendamment de la communication : grille CONT.
- Analyse de neuf logiciels selon la grille CONT.
- Etude des résultats : définition de la typologie CONT.

2 Etablissement d'une typologie du point de vue de la communication : typologie COM

- Construction d'une grille d'analyse a priori de la communication : grille COM.
- Analyse des neuf logiciels précédents selon la grille COM.
- Etude des résultats : définition de la typologie COM.

3 Comparaison des typologies CONT ET COM : recherche d'une typologie générale

Remarques

- Pour faciliter la lecture, nous regrouperons dans un fascicule, certains éléments de la partie I (les types des diverses typologies avec leurs notations, des tableaux de synthèse, ...).

- Les parties 1 et 2 de la partie I sont indépendantes : l'ordre ne concerne que l'exposé. Ceci permettra en particulier d'évaluer nos hypothèses.

- Le travail effectué dans la partie I de la recherche se veut "objectif", en ce sens qu'il essaie de ne pas introduire de critères de jugement.

- Les deux grilles d'analyse a priori CONT et COM sont "objectives" : elles analysent le logiciel sans émission d'hypothèses quant aux conditions, enjeux, et intérêts à l'emploi du produit en question dans une situation ayant pour objet d'enseignement un ou plusieurs concepts mathématiques. Les consignes données, le cas échéant, ne sont pas étudiées.

On ne s'intéresse pas non plus aux objectifs ni à l'organisation d'une situation construite avec le logiciel. En particulier, on ne se préoccupe pas du nombre d'enfants qui l'utilisent sur une machine.

Il s'agit, sans jugement d'intérêt ou d'adéquation, d'une analyse des potentialités du logiciel pour la construction d'une situation qui l'emploierait.

Il n'est pas fait d'étude de la qualité technique du logiciel.

2d Méthodologie relative à l'expression des enjeux didactiques (Partie II)

Dans la deuxième partie, nous examinerons la pertinence des critères d'analyse a priori et de la typologie, concernant la définition des tâches imparties à l'élève et la caractérisation du contexte de leur accomplissement.

Nous nous intéresserons ainsi nécessairement aux spécificités des logiciels pour l'enseignement, et de manière prospective, à la construction de nouveaux produits informatiques.

Nos outils seront :

- Les résultats de la première partie.
- Des observations que nous avons pu faire en situation

d'enseignement.

– Les outils de notre cadre théorique, la didactique des mathématiques.

Remarque

L'étude d'un glissement lors de la mise en œuvre d'un logiciel (logiciel d'un certain type utilisé dans un contexte attachable à un autre type) ne fait pas partie des objectifs de notre recherche.

PARTIE I

LOGICIELS POUVANT IMPLIQUER

DES ACTIVITES MATHÉMATIQUES

A L'ÉCOLE ÉLÉMENTAIRE :

TYPOLOGIE A PRIORI



TYPOLOGIE CONT

1- PREALABLE

Nous présenterons successivement :

- La grille CONT.

- Des commentaires explicatifs de cette grille.

Ces commentaires se retrouvent dans le fascicule d'accompagnement pour permettre une lecture accompagnée de la grille.

- L'étude comparative des résultats fournis par l'analyse a priori, selon la grille CONT, des neuf logiciels plus spécialement retenus pour la recherche.

Les analyses à proprement parler selon la grille CONT des neuf logiciels se trouvent en annexe III.

- La typologie a priori CONT.

Celle-ci est reprise dans le fascicule d'accompagnement.

D'autres logiciels que les neuf de référence sont classifiés en annexe IV selon la typologie générale, et donc en particulier selon la typologie CONT.

2- LA GRILLE CONT

2a Présentation de la grille CONT

La grille CONT s'intéresse aux niveaux d'enseignement, aux types de concepts impliqués par le logiciel, à l'organisation du logiciel par rapport à un concept, à l'action de l'enfant par rapport à l'activité (définition, gestion et validation de l'activité).

Les réponses à apporter sont de type binaire (oui/non), à moins que la question soit sans objet.

Niveau concerné

CP

CE

CM

Concepts mathématiques impliqués par le logiciel

Type arithmétique

Type géométrique

Le logiciel permet-il d'aborder plusieurs concepts de même type ?

Organisation du logiciel par rapport à un concept

Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?

Un nombre fini déterminé ?

Une infinité théorique ?

Les problèmes sont-ils nécessairement isomorphes ?

Oui Non

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?
- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

Oui Non

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message
 - . écriture
 - . graphisme
 - . son
- Contenu et cas d'envoi du message
 - . réponse exacte au problème
 - en cas d'erreur
 - toujours
 - . infirmation ou confirmation de la réponse
 - . autres éléments

L'enfant peut-il se sentir maître de la validation ?

- . Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?
- . Autrement ?

Oui Non

Zb A propos des rubriques de la grille CONT

Il est rappelé que la grille CONT propose une analyse a priori des logiciels du point de vue du contenu, sans émission d'hypothèses relatives aux caractéristiques et enjeux de situations qui emploieraient lesdits logiciels.

La grille CONT n'étudie pas la communication proposée à l'utilisateur avec le micro-ordinateur.

⇒ Rubrique : Niveau concerné

Cette rubrique précise le ou les niveaux de l'école élémentaire auxquels peut s'adresser le logiciel en référence aux concepts qu'il implique.

⇒ Rubrique : Concepts mathématiques impliqués par le logiciel

Cette rubrique s'intéresse au type (arithmétique ou géométrique) des principaux concepts mathématiques mobilisables par le logiciel.

L'implication de concepts peut intervenir :

- soit directement par la donnée de textes de problèmes par le logiciel.
- soit indirectement par la possibilité qu'offre le logiciel de construction d'activités, sans qu'il y ait proposition de textes explicites à l'écran.

⇒ Rubrique : Organisation du logiciel par rapport à un concept

Il s'agit d'examiner du point de vue du nombre et de la dépendance les problèmes que le logiciel pose ou peut permettre de poser.

"Isomorphe" signifie ici, données différentes, problème inchangé.

⇒ Rubrique : Action de l'enfant par rapport à l'activité

Sous-rubrique : Définition du contenu de l'activité

On dira qu'un des protagonistes définit le contenu de l'activité, si c'est lui qui décide du problème à traiter.

- Le logiciel peut le faire complètement ou partiellement, c'est-à-dire qu'il sera alors possible au maître ou à l'enfant de compléter la décision logicielle.

Notons que la question de la définition du contenu de l'activité est à examiner séparément de celle de la formulation des consignes ; en effet, la définition par le logiciel peut-être explicite (avec émission de textes) ou implicite (sans émission de textes).

- Le logiciel peut aussi être tel que son contenu fasse référence à un ou

plusieurs concepts sans pour autant qu'il décide des problèmes à traiter : l'enfant ou le maître peuvent alors décider et formuler l'activité à des degrés divers de prise en charge, suivant le logiciel.

Toutes ces questions concernent les possibilités de formulation de l'activité par le maître et la classe, avec des moyens oraux ou graphiques. On ne s'intéresse pas ici à des interventions sur le logiciel même.

Sous-rubrique : La donnée des consignes définissant le contenu de l'activité

On examine les éventuels supports de consignes attachés au logiciel.

La liberté d'émission de consignes orales, ou graphiques se déduit des réponses apportées à cette rubrique et à la précédente.

Sous-rubrique : La gestion de l'activité

Cette rubrique ne donne lieu à réponses que dans le cas où le logiciel définit les problèmes à traiter.

Signalons aussi que la rubrique suivante étudie à part la validation de l'activité.

On dira que "l'enfant doit répondre à des questions conceptuelles" si le logiciel propose des questions relatives à l'attendu conceptuel, qui soient intermédiaires entre la donnée du problème et sa solution.

On dira que "l'enfant peut avoir recours à une aide proposée par le logiciel", s'il existe une partie spécifique du logiciel, distincte du logiciel entier, à vocation d'aide pour la donnée de la réponse au problème traité. Pour préciser, cette partie est (ou pourrait être) optionnelle.

A titre d'exemple, l'aide pourrait consister en :

- la possibilité de modifier, de gérer l'écriture du problème.
- la possibilité de visualiser le problème, d'en avoir une autre expression.
- la possibilité de réaliser une action annexe propre à aider à la résolution du problème.

On rappelle que l'on ne juge pas ici de la pertinence de cette aide.

Sous-rubrique : Validation de l'activité par rapport à son contenu

On ne cherche pas à distinguer évaluation de validation, ni à cerner de façon précise le processus de validation.

Les questions ont seulement pour but de préciser dans quelle mesure l'enfant peut, ne serait-ce que se poser le problème de la validation de son

activité.

Par conséquent, on examine tout d'abord s'il y a un effet suite au traitement qu'a effectué l'enfant du problème posé : il y a un effet s'il se passe quelque chose ayant un rapport avec le traitement.

Remarquons que si le logiciel permet une intervention du maître ou de l'enfant dans la définition des problèmes à traiter, on peut avoir affaire à des consignes au niveau de l'objet à produire, mais aussi au niveau de conditions à respecter pour l'obtenir ; dans ce cas, il peut y avoir effet au niveau de l'objet, mais éventuellement non au niveau de la démarche : pour ces logiciels, on se contentera de répondre à la question au niveau de l'objet à produire (ce qui peut être le contenu complet de l'activité). On restera conscient de la lacune éventuelle au niveau de la validation du respect des conditions : ce point sera repris par la formulation "se sentir maître de la validation".

Si le logiciel ne permet pas une intervention extérieure dans la définition des problèmes à traiter, la question de l'effet qui permet de valider est posée pour le contenu complet de l'activité tel qu'il est défini par le logiciel.

En cas de réponse positive à la question concernant l'effet, on recherche par des questions qui ne sont pas exclusives sur qui repose la validation.

Par rapport à l'effet qui permet de valider, le problème est : qui produit cet effet ?

— La machine ?

Elle peut le faire sans pour autant donner de réponse au problème traité (infirmité ou confirmation de la réponse seulement).

La machine peut aussi apporter des éléments en fournissant une autre expression de la réponse de l'enfant.

Il peut exister d'autres données attachées au message de validation (en particulier, indications relatives au temps, envoi de scores, ...) : elles pondèrent la validation, mais ne sont pas la validation à strictement parler. Leur présence est signalée par "autres éléments" et sera reprise par la grille COM.

— L'enfant ?

Expliquons la formulation "se sentir maître de la validation" :

L'élève peut penser valider, et en fait n'être pas en mesure de le faire réellement soit au niveau de l'objet, — c'est ainsi que des enfants de CE1 ont validé comme carré un graphisme produit sous LOGO par une série d'instructions où les rotations étaient de 89° —, soit au niveau des contraintes de production de l'objet.

Il peut aussi être tenté de valider une réponse à un autre problème que celui initialement posé (production d'un objet intermédiaire par exemple).

La visualisation de la production (écriture, graphisme) semble être le point d'ancrage de la validation par l'enfant ; la rubrique "autrement" est ouverte pour tenir compte d'éventuels autres moyens (sonores, mobiles, ...).

3— ETUDE COMPARATIVE DES RESULTATS FOURNIS PAR L'ANALYSE A PRIORI DE LOGICIELS SELON LA GRILLE CONT

Les pages suivantes décrivent l'élaboration de la typologie CONT, en référence à l'analyse de neuf logiciels.

Ce travail a été effectué en 1983 : les logiciels cités ont été examinés à cette date, dans leur état de réalisation informatique. Ils sont de provenances diverses : on trouve des logiciels commercialisés en 1983 (LOGO, PICTOR), des prototypes de logiciels ultérieurement commercialisés (FRISE, DIVIS), des produits construits par des élèves-institutrices en formation initiale (MOSAIQUE, PUCE), des produits mis au point à fin de recherche (SIMUL, RECT, TACHE).

Le lecteur pourra consulter en annexe I un descriptif sommaire de ces logiciels, et en annexe III l'analyse de chaque logiciel.

Les éléments de réponses figurant dans les tableaux qui suivent proviennent de l'analyse rapportée en annexe III.

3a Etude comparative des résultats des rubriques "Niveau concerné", "Concepts mathématiques impliqués par le logiciel", "Organisation du logiciel par rapport à un concept"

On peut établir le tableau suivant :

	Puce	Divis	Rect	Tache	Simul	Frise	Mos	Pictor	Logo
Tout niveau					X	X	X	X	X
Concept arithmétique	X	X	X	X					X
Concept géométrique				(X)	X	X	X	X	X
Plusieurs concepts d'un même type						X	X	X	X
Plusieurs problèmes sur un concept	X	X	X	X	X	X	X	X	X
Nombre fini de problèmes	X	X	X	X	X	X	X	X	X
Infinité de problèmes						X	X	X	X
Problèmes isomorphes	X	X	X	X	X				

On remarque que tous les logiciels permettent de traiter un nombre fini, supérieur à 1, de problèmes portant sur un concept : ces deux critères sont écartés pour l'établissement de types et ne sont retenus que pour mémoire.

Les logiciels se répartissent en quatre groupes :

- Groupe 1 : PUCE DIVIS RECT TACHE

Tache peut être compris dans ce groupe. En effet l'aspect géométrique est secondaire face au problème de dénombrement sur lequel est posée la question.

- Groupe 2 : FRISE MOSAIQUE PICTOR
- Groupe 3 : LOGO
- Groupe 4 : SIMUL

• EXAMEN DES GROUPES DEGAGES

Les deux principaux groupes sont :

Le groupe 1 : logiciels

- sur certains niveaux,
- impliquant un concept,
- arithmétique,
- pour un nombre fini de problèmes,
- isomorphes.

Le groupe 2 : logiciels

- tout niveau,
- impliquant plusieurs concepts,
- géométriques,
- pour une infinité de problèmes,
- non isomorphes

Ces deux groupes sont opposés : chaque caractéristique de l'un trouve sa négation dans l'autre.

Quelques questions à propos des groupes 1 et 2 :

Les logiciels du groupe 1 analysés font intervenir des concepts arithmétiques, tandis que ceux du groupe 2 impliquent des concepts géométriques. Le type de concept traité par un logiciel en détermine-t-il les caractéristiques ? Quel rapport y-a-t-il avec les spécificités de la machine ?

Le groupe 2 semble plus riche de possibilités : les concepts géométriques permettraient-ils de mieux exploiter un ordinateur ?

Ainsi, le groupe 2 s'adresse à tous les niveaux de l'école : d'où provient cette apparente ouverture ? Nous avancerons que le domaine géométrique permet de conduire des activités qui, bien que différentes, peuvent être basées sur le même matériel pour tous les niveaux de l'école élémentaire. Ceci nous semble être la raison de la spécificité "tout niveau" du groupe 2. C'est pourquoi nous ne retiendrons pas ce critère comme véritablement pertinent.

Toutefois, cette caractéristique étant écartée, la question reste : les diverses possibilités offertes par chaque groupe sont-elles liées à la structure du logiciel ou au type de concept ?

Le groupe 3

Il possède :

- les mêmes caractéristiques que le groupe 2,
- et, en outre, la possibilité de traiter de concepts arithmétiques et géométriques.

Nous retiendrons cette différence avec le groupe 2, mais, observant, que l'on ne fera pas intervenir simultanément les deux types de concepts avec le logiciel, nous rattacherons le groupe 3 au groupe 2.

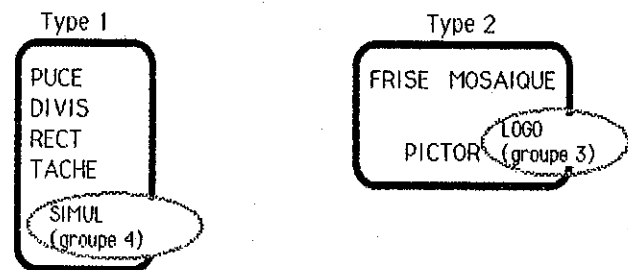
Le groupe 4

Il peut-être rapproché du groupe 2 pour les critères "tout niveau" et "concept géométrique".

Suite à l'abandon du critère "tout niveau" que nous avons proposé, il s'inscrit dans le groupe 1, et affirme ainsi l'évidente possibilité d'activités à caractère géométrique pour celui-ci.

• MISE AU POINT

Cette première étude permet le classement des logiciels étudiés selon deux types correspondants aux groupes 1 et 2.



Une première expression de la typologie recherchée est en conséquence la suivante :

Pour un niveau donné d'enseignement, les logiciels se répartissent selon deux types :

- Le type 1

Logiciels impliquant un concept arithmétique ou géométrique pour un nombre fini de problèmes isomorphes.

- Le type 2

Logiciels impliquant plusieurs concepts arithmétiques ou géométriques pour une infinité de problèmes non isomorphes sur chaque concept.

• QUESTIONS

- L'implication d'un concept de genre géométrique est assurée par "SIMUL" en type 1. Elle pourrait l'être par d'autres logiciels.

A l'inverse, pour le type 2 tel qu'il est à ce stade défini, nous ne connaissons pas au moment où nous écrivons d'autres représentants que LOGO pour proposer des concepts de type arithmétique.

Or, LOGO est essentiellement utilisé pour sa partie géométrique.

Ainsi, le genre de concept, à savoir "géométrique" semble déterminant en type 2.

Est-il possible qu'il en soit autrement ?

- L'isomorphie des problèmes va de pair avec le nombre fini de ceux-ci (de même la non-isomorphie avec l'infinité théorique). Peut-on envisager des logiciels qui ne présentent pas cette corrélation ? Cela ne semble pas impossible.

3b Etude comparative des résultats de la rubrique "Action de l'enfant par rapport à l'activité"

⇒ SOUS-RUBRIQUE "DEFINITION DU CONTENU DE L'ACTIVITE"

Nous analyserons tout d'abord les résultats de la sous-rubrique "la définition du contenu de l'activité" ; elle est en effet la question centrale qui tente de déterminer quel degré de liberté pourra exister pour l'enfant, le maître ou la classe à ce propos, s'il est décidé d'utiliser le logiciel pour la construction d'une situation.

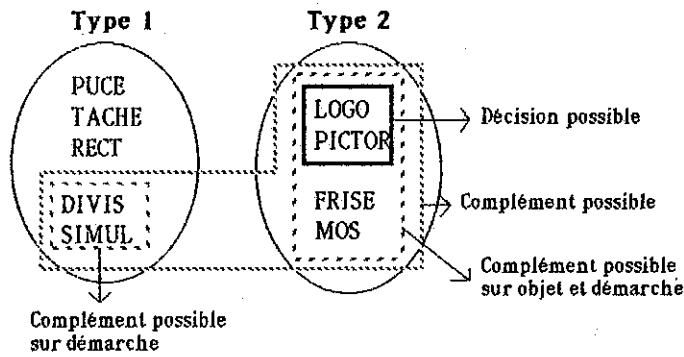
Remarque : la formulation "le logiciel décide-t-il des problèmes à traiter" est volontairement anthropomorphe.

	Puce	Divis	Rect	Tache	Simul	Frise	Mos	Pictor	Logo
Le logiciel décide des problèmes	X	X	X	X	X	X	X		
Le logiciel ne décide pas					X	X	X	X	X
Complément de définition possible		X		X	X	X	X	X	X
Sur l'objet de l'activité					X	X	X	X	X
Sur la démarche		X		X	X	X	X	X	X

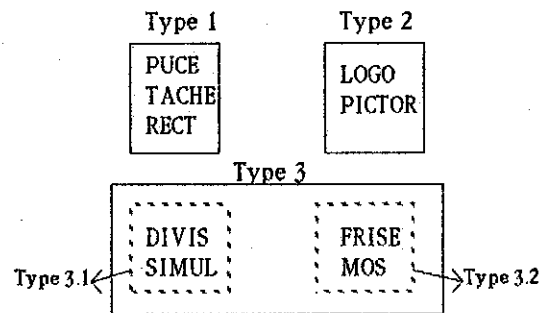
L'examen des deux premières lignes conduit à détacher LOGO et PICTOR des autres logiciels.

Dans un deuxième temps, l'examen de la troisième ligne permet de rapprocher DIVIS, SIMUL, FRISE et MOSAIQUE de LOGO et PICTOR : ils autorisent des compléments de définition, mais ceci à des degrés divers, et seront corrélativement plus ou moins éloignés de LOGO et PICTOR ; les deux dernières lignes précisent ce fait.

On est donc amené à remodeler les types dégagés en 3a :



Ce qui permet de proposer trois types,



• MISE AU POINT

Pour l'analyse comparative précédente, il existait une distance entre LOGO et PICTOR (possibilité d'impliquer, avec LOGO, des concepts géométriques ou arithmétiques) que nous n'avions d'ailleurs pas retenu comme réellement pertinente.

Pour les critères étudiés ici, il n'y a plus de différenciation entre ces

deux logiciels.

Les types 1 et 2 sont toujours opposés. Il n'y a plus de logiciels à caractère géométrique en type 1, mais leur existence dans ce type pourrait être attestée par ailleurs.

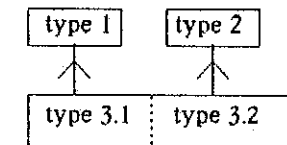
Le type 3 est "mixte" : il possède des caractéristiques du type 1 ou du type 2.

Par rapport aux types 1 et 2 dégagés ici, le type 3 peut être séparé en deux sous-types qui expriment la distance avec ceux-ci :

- Le sous-type 3.1 : en provenance du type 1 précédent, éloigné du type 2 par le fait que le complément de définition ne peut porter sur l'objet du problème.

- Le sous-type 3.2 : en provenance du type 2 précédent, proche du nouveau type 2 : si le logiciel décide du problème à traiter, il permet néanmoins un complément de définition sur l'objet du problème ; la décision logicielle est donc toute relative. En fait, l'objet du problème peut être défini par la classe (maître-enfants) comme c'est le cas en type 2, mais sur un champ plus restreint. On peut d'ailleurs faire remarquer que pour le type 2, si la décision est possible, elle se fait néanmoins sur un domaine donné, celui que permet de traiter le logiciel : on ne peut pas faire n'importe quoi avec LOGO ou PICTOR.

On peut schématiser l'éloignement des divers types ainsi, une flèche désignant la tendance d'un type vers un autre.



En synthèse, nous présenterons maintenant la **typologie recherchée en trois types** :

type 1 : logiciels impliquant 1 concept arithmétique ou géométrique pour un nombre fini de problèmes isomorphes et définissant complètement le contenu de l'activité.

type 2 : logiciels impliquant plusieurs concepts arithmétiques¹ ou géométriques pour une infinité de problèmes non isomorphes sur chaque concept et laissant à autrui la définition complète du contenu de l'activité (sur la base des éléments qui déterminent le logiciel).

1. Le critère "arithmétique" n'est assuré que par LOGO.

type 3 : logiciels définissant partiellement le contenu de l'activité.

type 3.1 : impliquant 1 concept arithmétique ou géométrique pour un nombre fini de problèmes isomorphes.

Le complément de définition porte sur la démarche.

type 3.2 : impliquant plusieurs concepts géométriques pour une infinité de problèmes non isomorphes sur chaque concept.

Le complément de définition porte sur l'objet.

⇒ SOUS-RUBRIQUE "LA DONNEE DES CONSIGNES DEFINISSANT LE CONTENU DE L'ACTIVITE"

	Type1			Type3				Type2	
	Puce	Rect	Tache	Divis	Simul	Frise	Mos	Pictor	Logo
Il y a des consignes	X			X		(X)	X		
Sur l'écran	X			X		(X)	X		
Sur une documentation									
Elles sont données en une fois	(X)			X		(X)	X		

Cette sous-rubrique permet de préciser les trois types présentés auparavant :

A propos de l'expression des consignes dans les divers types de logiciels

Le type 2 : L'absence de consignes est cohérente avec la définition du type et est caractéristique de celui-ci.

L'individu décide du problème et émet des consignes, ce qui pourra se faire de diverses façons (oralement, sur document ...).

Le type 1 : Compte-tenu de sa définition on aurait pu penser que la présence de consignes à l'écran était une obligation. "PUCE" (et d'autres logiciels non étudiés ici) le confirme. Nous avancerons que la présence de consignes est signifi-

cative de ce type :

la machine décide complètement du contenu et émet les consignes. Aucune place n'est laissée à l'individu pour cette phase de l'activité.

Certains logiciels dérogent à cette règle : ils empruntent la caractéristique "absence de consignes" au type 2.

Notons qu'il s'agit des logiciels conçus dans le cadre de l'I.R.E.M à fin de recherche. (RECT-TACHE). Dans ce cas, la machine décide complètement du contenu de l'activité, mais un degré de liberté est laissé à l'individu pour l'expression des consignes.

Le type 3 : Pour l'expression des consignes, certains logiciels se rattachent au type 1 (DIVIS, MOSAÏQUE surtout, FRISE dans une moindre mesure), d'autres au type 2 (SIMUL, FRISE dans une moindre mesure).

Nous remarquerons que si le type 3.1 peut justifier la présence de consignes à l'écran (avec la perte de liberté d'expression) en raison de la définition complète de l'objet de l'activité par le logiciel, le type 3.2 ne devrait pas rendre celle-ci nécessaire.

On constate d'ailleurs que les consignes de MOSAÏQUE et FRISE sont minimum. Elles ne concernent pas directement une activité mathématique explicite.

Les consignes spécifiques de FRISE ("choisis ta machine à friser" ; "Dessine sur le bleu clair") ne sont pas perçues comme définissant l'activité par l'utilisateur : on peut observer qu'en l'absence d'une intervention extérieure, celui-ci s'interroge sur l'objet de l'activité.

Enfin, de façon générale, les logiciels qui émettent des consignes le font sur l'écran et jamais par leur environnement (document graphique). Une autre proposition pourrait être adoptée tout spécialement en type 3.2.

A propos du moment de donnée de consignes dans les divers types de logiciels

Lorsque des consignes sont données par le logiciel, elles le sont en une fois, avec une restriction pour PUCE qui émet des consignes en cours d'activité (proposition d'aides diverses).

Notons qu'il existe des logiciels qui émettent des consignes intermédiaires.

• MISE AU POINT

L'expression des consignes par l'individu est possible quel que soit le type du logiciel.

Elle est néanmoins caractéristique du type 2.

Le type 1 présente généralement les consignes à l'écran. Elles sont données soit en une fois, soit au fur et à mesure.

Certains logiciels du type 1 dérogent à la règle pour l'expression des consignes et se rattachent au type 2 pour ce critère.

Les logiciels du type 3 se rattachent au type 1 ou au type 2 pour l'expression des consignes.

⇒ SOUS-RUBRIQUE "LA GESTION DE L'ACTIVITE"

	Puce	Divis	Rect	Tache	Simul	Frise	Mos	Pictor	Logo
Choix d'un type de problèmes parmi plusieurs									
Choix d'un niveau		X							
Cursus obligatoire de plusieurs problèmes	X	X	X	X	X				
Questions conceptuelles intermédiaires									
Aide	X			X					

Il s'agit d'évaluer le degré d'intervention du logiciel dans la gestion de l'activité, par des propositions directes (définies par les questions de cette rubrique). Remarquons que l'on ne s'intéresse pas ici à l'intervention due à la communication avec la machine.

Lorsque le logiciel est de type 2, l'influence du logiciel sur la gestion de l'activité ne se présente pas en termes analysables par les questions de cette rubrique : le logiciel n'intervient pas par des propositions directes.

FRISE et MOSAIQUE n'interviennent pas non plus. Par conséquent, les réponses apportées à l'ensemble des questions, et plus spécialement à celle du

"cursus obligatoire" permettent de retrouver les deux types dégagés initialement (type 1 : PUCE DIVIS RECT TACHE SIMUL ; type 2 : FRISE MOSAIQUE PICTOR LOGO).

Pour le groupe des logiciels sur lesquels porte principalement la rubrique "gestion de l'activité" (types 1 et 3), on ne retrouve pas nettement la différenciation 1 - 3 ; on ne peut pas non plus en dégager une nouvelle. Tout se passe comme si la sous-rubrique "définition du contenu de l'activité" était déterminante pour la distinction plus affinée des logiciels, et comme si la sous-rubrique "gestion de l'activité" ne faisait qu'apporter des précisions relatives aux types 1 et 3, que nous allons maintenant évoquer.

Pour les types 1 et 3, il n'y a pas de questions conceptuelles intermédiaires.

Elles sont concevables en type 1 (des logiciels peuvent l'attester) et en type 3.1, plus difficilement en 3.2 en raison du complément de définition possible sur l'objet de l'activité.

Nous qualifierons les types 1 et 3 pour la gestion de l'activité ainsi :

Type 1 : Le degré d'intervention se mesure pour les logiciels étudiés à la proposition d'un cursus obligatoire de problèmes, et à l'éventuelle proposition d'aides.

Des logiciels non retenus ici rendraient compte des autres options d'intervention.

Type 3 :

Sous-type 3.2 : Il est bien déterminé, comme ne proposant rien.

Ainsi l'ensemble des logiciels du sous-type 3.2 se situe à proximité du type 2 en raison d'une caractéristique de non-intervention directe dans la gestion de l'activité.

Sous-type 3.1 :

Il propose un cursus de problèmes obligatoire.

Il n'y a pas d'autre intervention pour SIMUL (tendance vers le type 2) ; DIVIS intervient en proposant des niveaux de difficulté (tendance vers le type 1)

• MISE AU POINT

En types 1 et 3.1, il y a intervention du logiciel dans la gestion de l'activité par la donnée d'un cursus obligatoire de problèmes et le cas échéant, par d'autres propositions.

En types 2 et 3.2, il n'y a pas d'intervention du logiciel dans la gestion de l'activité par des propositions directes.

• QUESTIONS

Le nombre de propositions faites par le logiciel peut-il constituer une mesure de son degré d'intervention dans l'activité ?

Peut-on, sur cette base, reconnaître dans un logiciel de type 3 une tendance vers le type 1 ou le type 2 ?

Quel est le type d'intervention du logiciel, lorsque celui-ci ne fait pas de propositions directes ?

Quels sont les enjeux des divers types d'intervention ?

Peut-on qualifier la tendance d'un logiciel uniquement par le décompte de ses interventions directes ?

Selon le type de logiciel utilisé dans une situation d'enseignement, quel rôle a l'ordinateur ?

3c Etude comparative des résultats de la sous-rubrique "La validation de l'activité par rapport à son contenu"

La validation fait partie de la gestion de l'activité, mais il a été choisi de l'étudier à part pour savoir si elle est déterminante de types de logiciels.

⇒ QUESTIONS CONCERNANT LES DETENTEURS DE LA VALIDATION

	Type1			Type3				Type2	
	Rect	Puce	Tache	Divis	Simul	Frise	Mos	Pictor	Logo
Y-a-t-il un effet ?		X	X	X	X	X	X	X	X
"La machine valide"		X	X	X					
"L'enfant valide"				(X)	X	X	X	X	X

Sous-rubrique : "Y-a-t-il un effet" ?

Pour tous les logiciels, sauf RECT, lorsque l'enfant a terminé l'activité, il y a un effet relatif à celle-ci. Les formes de cet effet seront précisées dans les lignes qui suivent, et lors de l'étude de la communication.

Pour RECT (produit conçu à fin de recherche), nous émettrons au cours de la deuxième partie de la recherche des hypothèses quant à l'incidence sur la validation de l'absence d'effets. En particulier, nous tenterons de cerner ce que devient la validation.

Généralement, l'usage d'un logiciel conduit à un effet qui devrait engager l'enfant à se poser le problème de la validation. Nous examinerons au cours de la partie II de la recherche en quoi celle-ci consiste, si elle a réellement lieu, si elle se limite ou non à une évaluation. Au stade où nous sommes, nous dirons que les logiciels ne sont pas différenciés en ce qui concerne la mobilisation de l'utilisateur pour une validation de son activité.

Sous-rubrique : "La machine fait-elle la validation pour l'enfant ?"
"L'enfant peut-il se sentir maître de la validation ?"

Les réponses permettent de classer les logiciels suivant celui de la machine ou de l'utilisateur qui détient potentiellement la validation.

Il est possible de constituer deux groupes, complétés d'un logiciel isolé RECT.

Les groupes sont :

Le groupe 1 : PUCE, DIVIS, TACHE

Le groupe 2 : SIMUL, FRISE, MOSAÏQUE, PICTOR, LOGO

Nous ferons deux remarques.

Si RECT a une particularité qui ne sera pas oubliée, il serait possible qu'il donne lieu aux mêmes réponses que les logiciels du groupe 1 (en le complétant d'une analyse de réponses et d'envois de message), mais pas à celles des logiciels du groupe 2.

D'autre part, DIVIS pourrait laisser l'enfant maître de la validation. En effet, rappelons que le message envoyé par le logiciel est superflu eu égard à l'activité d'obtention du quotient et du reste : il n'est pertinent que pour une optimisation de leurs productions.

Ceci permet de dire qu'il y a confirmation :

– de l'existence du type 3 précédemment défini, par un nouveau critère "l'enfant peut se sentir maître de la validation"

– de la proximité du type 3 au type 2 par le critère "l'enfant peut se sentir maître de la validation".

Toutefois, nous n'oublierons pas que certains logiciels du sous-type 3.1 (DIVIS par exemple) peuvent se caractériser en type 1 pour la validation.

– de l'existence du type 1 précédemment défini, par un nouveau critère "la machine fait la validation".

La partie II de la recherche tentera de cerner le statut de cette validation, comme ses conséquences sur l'activité de l'utilisateur.

⇨ QUESTIONS CONCERNANT LES BASES DE LA VALIDATION

Nous avons vu que pour les logiciels des types 2 et 3, il n'y a pas de distinction : l'enfant peut se sentir maître de la validation sur la base de la production qu'il a fait faire à la machine.

A l'inverse, pour les logiciels de type 1, l'enfant constate un message du logiciel ; on peut néanmoins établir des différences précisant l'intervention du logiciel.

Support du message de validation

	Puce	Divis	Tache
Écriture	X	X	X
Graphisme		(X)	X
Son	X	X	

Le message émis pour la validation comporte toujours une écriture, mais s'exprime aussi par d'autres supports : au total,

- 2 supports pour PUCE et TACHE
- quasiment 3 supports pour DIVIS (effet graphique en fin d'activité : réservoir vide).

Le degré d'intervention est donc plus ou moins grand.

Quels sont les enjeux de l'expression sur les divers supports ?

Contenu et cas d'envoi du message de validation

	Puce	Divis	Tache
Réponse exacte		X toujours	X si erreur
Autre expression de la réponse de l'enfant		X toujours	X si erreur
Infirmation/ Confirmation	X	X	X
Autres éléments		X (Score)	X Score

Le message comporte toujours infirmation ou confirmation de l'activité ; toutefois, il existe, le cas échéant, d'autres interventions du logiciel sur la gestion de l'activité au moment de la validation.

DIVIS présente le plus grand degré d'intervention, puis vient TACHE, et enfin PUCE.

Quels sont les enjeux de l'emploi de ces diverses données dans un message de validation ?

• MISE AU POINT

L'examen de la validation de l'activité ne remet pas en cause la typologie en trois types présentée auparavant.

Il apporte un complément d'information :

- En type 2,3, l'enfant peut se sentir maître de la validation par visualisation de la production qu'il a fait faire à la machine.
- En type 1 : La machine "fait la validation" à divers degrés d'intervention.

• QUESTIONS

L'enfant se pose-t-il le problème de la validation dans une situation qui utilise un logiciel ?

Qu'advient-il de celle -ci ?

Peut-il y avoir réellement validation par l'enfant ?

Que valide-t-il ?

4- DEFINITION DE LA TYPOLOGIE CONT

4a Descriptif de la typologie CONT

L'étude des résultats fournis par l'analyse a priori des neuf logiciels selon la grille CONT permet de proposer une typologie en trois types.

Celle-ci présente deux pôles (le type 1 et le type 2) et un type intermédiaire (le type 3).

Les trois types de la typologie CONT se caractérisent ainsi :

Le type 1

Logiciels impliquant un concept arithmétique ou géométrique concernant un nombre fini de problèmes isomorphes,

- définissant complètement le contenu de l'activité et l'exprimant à l'écran par des consignes données en une fois ou au fur et à mesure,
- intervenant dans la gestion de l'activité par des propositions directes,
- effectuant la validation de l'activité.

Le type 2

Logiciels impliquant une infinité de problèmes non isomorphes concernant plusieurs concepts arithmétiques ou géométriques,

- laissant à autrui la définition complète du contenu de l'activité et son expression,
- n'intervenant pas dans la gestion de l'activité par des propositions directes,
- permettant à l'enfant de se sentir maître de la validation.

Le type 3

Ce type comporte deux sous-types : les sous-types 3.1 et 3.2.

Caractéristiques générales du type 3

Logiciels définissant partiellement le contenu de l'activité,

- l'exprimant à l'écran par des consignes ou laissant autrui l'exprimer,
- permettant à l'enfant de se sentir maître de la validation.

Caractéristiques spécifiques au sous-type 3.1

Logiciels impliquant un nombre fini de problèmes isomorphes sur un concept arithmétique ou géométrique.

Le complément de définition est possible sur la démarche.

Le logiciel intervient dans la gestion de l'activité par des propositions directes.

Caractéristiques spécifiques au sous-type 3.2

Logiciels impliquant une infinité de problèmes non isomorphes concernant plusieurs concepts géométriques.

Le complément de définition est possible sur l'objet de l'activité (et a fortiori sur la démarche).

Le logiciel n'intervient pas dans la gestion de l'activité par des propositions directes.

Rappel : Les neuf logiciels analysés se répartissent ainsi :

Type 1 : PUCE, RECT, TACHE

Type 2 : LOGO, PICTOR

Type 3 : Sous-type 3.1 : DIVIS, SIMUL

Sous-type 3.2 : FRISE, MOSAIQUE

En outre, d'autres logiciels sont classés en annexe IV selon la typologie générale, et donc en particulier selon la typologie CONT.

4b Quelques remarques

- Les types 1 et 2 sont opposés. De plus, pour les logiciels dont nous avons eu connaissance, on peut noter que les logiciels du type 2 font intervenir essentiellement des concepts géométriques.

- Il y a proximité des types 2 et 3 ; si cependant le sous-type 3.2 est le plus proche du type 2, le sous-type 3.1 a plus de points communs avec le type 1.

- Le classement d'un logiciel dans un type peut s'avérer relatif : des logiciels relevant du type 1 empruntent certaines caractéristiques au type 2 ; des logiciels du type 3 présentent des éléments spécifiques au type 1 ou 2.

4c Quelques questions

Nous rappelons que nous apporterons, au cours de la partie II de la recherche, des éléments de réponses aux questions que nous allons maintenant poser.

- Les faits suivants sont-ils contingents ?
 - . type 2 et concept géométrique essentiellement.
 - . nombre fini de problèmes portant sur un concept et isomorphie de ceux-ci.
 - . type 1 et définition du contenu de l'activité par le logiciel.
 - . type 2 et non définition du contenu de l'activité par le logiciel.
 - . type 1 et émission de consignes par le logiciel.
 - . type 2 et non émission de consignes par le logiciel.

- Le type 3 est marqué par le type 1 ou le type 2, mais tous les logiciels de ce type présentent une certaine communauté (essentiellement sur la possibilité de définition partielle de l'activité par l'utilisateur) : peut-on expliquer ce fait ?

- La mesure du degré d'intervention du logiciel dans l'activité peut-elle s'effectuer par le nombre des propositions qu'il fait pour la gestion de l'activité (y compris la validation de celle-ci) ? Est-ce une mesure valable ? suffisante ?

Que dire alors de l'intervention des logiciels du type 2 dans l'activité ?

- Qu'advient-il de la validation dans les divers types ?

Quel est le rôle des divers supports impartis aux messages de validation, le cas échéant ?

Lorsque l'enfant se sent maître de la validation, que valide-t-il ?

- Les différences entre les types de CONT recouvrent-elles d'autres différences ?

- Il est rappelé que la typologie CONT ne prétend pas, à ce stade, être un instrument de mesure de l'intérêt des logiciels. Quels sont les enjeux de situations utilisant des logiciels des divers types ?

TYPLOGIE COM

1- PREALABLE

Nous procéderons comme pour la typologie CONT, et présenterons successivement :

- La grille COM.

- Des commentaires explicatifs de cette grille.

Ces commentaires se retrouvent dans le fascicule d'accompagnement pour faciliter la lecture de la grille.

- L'étude comparative des résultats fournis par l'analyse a priori, selon la grille COM, des neuf logiciels plus spécialement retenus pour la recherche.

Les analyses à proprement parler selon la grille COM des neuf logiciels se trouvent en annexe III.

- La typologie a priori COM.

Celle-ci est reprise dans le fascicule d'accompagnement.

D'autres logiciels que les neuf de référence sont classifiés en annexe IV selon la typologie générale, et donc en particulier selon la typologie COM.

2- LA GRILLE COM

2a Présentation de la grille COM

La grille COM, que nous présentons à la page suivante, recourt à la notion d'«instruction» que nous avons jugé propre à l'étude de la communication de l'utilisateur avec le micro-ordinateur. Elle s'intéresse également à la nature, l'organisation, l'évaluation, le statut des entrées de l'utilisateur.

Les réponses à apporter sont de type binaire (oui/non), à moins que la question soit sans objet.

**Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur**

Oui Non

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

Sur un document ?

Sur un autre type de support ?

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

Certaines sont-elles valides sur une partie du logiciel ?

Certaines sont-elles valides à un seul endroit du logiciel ?

Les entrées-utilisateur

Oui Non

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

Evaluation

Caractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type 1

Y-a-t-il exécution avec production d'objets commandés par l'entrée-utilisateur ?

La sous-rubrique "caractérisation" invite à classer les instructions qu'elle examine en quatre types (a - b - c - d).

Précisons ces quatre types :

Instructions que l'utilisateur peut recevoir de la machine	Instructions que l'utilisateur peut donner à la machine
type a : instructions de définition de l'activité	type d : instructions de production
type b : instructions questions à l'utilisateur par rapport au problème qu'il traite	type c : instructions de parcours du logiciel - type branchement dans le logiciel
type c : instructions de parcours du logiciel - type repérage dans le logiciel	

• Les instructions de type c (instructions de parcours du logiciel) sont de deux sortes :

- type repérage dans le logiciel : elles servent à préciser les entrées que l'utilisateur pourra employer par la suite.
- type branchement dans le logiciel : sans implication réelle dans une production ultérieure, elles ne font que donner accès à certaines parties du logiciel. Elles constituent l'expression d'un choix (de niveau de difficulté, par exemple) et ne sont pas significatives pour la suite de l'activité.

• Les instructions de type d (instructions de production) sont constituées :

- des instructions qui permettent à l'utilisateur de faire produire immédiatement la machine.
- des instructions qui sont en fait des instructions de branchement, mais qui ont une implication directe dans une production : elles sont significatives dans celle-ci.

QUELQUES REMARQUES, EXEMPLES, CONTRE-EXEMPLES

Certaines instructions se présentent sous forme de phrases.

Voici deux exemples :

- "Tu vas faire des divisions" (instruction de type a).

- "Choisis un niveau (A-B)" (instruction de type c - repérage dans le logiciel).

D'autres instructions se présentent sous forme de codages.

Toutefois, tous les codages ne sont pas des instructions au sens défini pour cette recherche.

Examinons les codages par lettres, nombres et expression "oui/non".

Il convient de distinguer :

- les codages qui traduisent des données brutes du logiciel.

Ils seront retenus comme instructions.

On trouve :

- des codages locaux, sans rapport explicite entre signifiant et signifié concernés : ainsi, le mot "oui" ou le chiffre 1 peuvent coder des faits différents sur l'ensemble du logiciel.

On n'oubliera pas qu'ils s'utilisent généralement en réponse à une demande de la machine.

Ils concernent essentiellement des choix de situation.

Exemples :

- Choisis un niveau (A-B)
 - Veux-tu faire des additions ? (oui-non)
- "A-B-oui-non" sont ici des instructions de type c - branchement dans le logiciel.

- des codages valides pendant la totalité du logiciel. Un code traduit une donnée brute du logiciel et une seule.

Il y a souvent un rapport explicite entre signifiant et signifié (initiale par exemple). Ils sont généralement réalisés par lettres et ne s'utilisent pas nécessairement en réponse à une demande de la machine.

Exemples :

- Pour donner la réponse, tape R
 - Pour que la machine dessine une ligne, tape L
- "R" est une instruction de type c - branchement dans le logiciel ;
"L" est une instruction de type d.

Remarquons qu'il peut exister des codages de données brutes valides sur tout le logiciel, réalisés par autre chose qu'une lettre (graphisme sur écran - cas des instructions LOGO par exemple).

- les codages qui traduisent des constructions personnelles de l'utilisateur, indépendamment des moyens logiciels.

Ils ne seront pas retenus comme instructions.

Exemples :

- Le nombre 21 est-il :
 - A - multiple de 7 ?
 - B - multiple de 5 ?
 Donne la réponse (A ou B).
 - Pour résoudre ce problème, faut-il faire une addition ? (oui - non)
- "A-B-oui-non" ne sont pas dans ces cas des instructions.
 D'une façon générale, les codages de réponses des logiciels QCM ne sont pas des instructions au sens défini pour cette recherche.

CAS PARTICULIERS

- On appellera «invite-utilisateur» le signal qui fait que l'utilisateur sait qu'il peut effectuer une entrée.

Il s'agit généralement d'un symbole ou du curseur, éventuellement de l'attente de la machine suite à ce qui précède (invite implicite).

L'instruction qui précède le cas échéant l'invite à proprement parler n'est pas incluse dans l'invite.

D'après le sens donné au mot instruction, l'invite constitue une instruction. Attendu qu'elle devrait s'exprimer de la même façon tout au long du logiciel, elle ne sera pas prise en considération.

- Une instruction produite par la machine demandant le nom de l'utilisateur (ou d'autres renseignements sur son compte) ne sera pas retenue lors de l'analyse d'un logiciel : bien que significative pour les rapports machine-utilisateur, elle n'entre pas dans l'activité proprement dite. L'entrée-utilisateur qui lui correspond ne sera pas non plus étudiée.

Sous-rubrique : Support des instructions

Ces supports peuvent être variés. Certains périphériques d'entrée (crayon-optique, souris ...) peuvent rendre le support écran nécessaire.

Sous-rubrique : Validité des instructions

Les instructions de types a, b et c-repérage sont données par le logiciel, et généralement valides une fois, ou au mieux périodiquement sur le logiciel.

Les instructions de types d et c-branchement peuvent être données par l'utilisateur à la machine (elles peuvent être des entrées-utilisateur) et l'on constate qu'elles peuvent avoir un plus grand domaine de validité : la sous-rubrique "validité des instructions" étudie la validité des instructions qui peuvent être des entrées-utilisateur.

De façon plus précise, on distingue trois domaines de validité :

- toutes les instructions sont valides sur tout le logiciel.

- Certaines instructions sont valides sur une partie du logiciel.
 La partie en question peut être constituée d'une réunion de diverses parties disjointes.

- Certaines instructions sont valides à un seul endroit du logiciel.

Ainsi, l'examen de la validité des instructions renseigne sur la possibilité qu'a l'utilisateur d'effectuer une entrée avec des instructions diverses.

Lorsqu'un même problème se répète plusieurs fois, il ne faut répondre aux questions de cette sous-rubrique que pour un exemplaire du problème.

⇒ Rubrique : Les entrées-utilisateur

Dans cette partie, le terme écriture s'entend au sens large, c'est-à-dire qu'il s'agit d'une écriture réalisée avec un périphérique d'entrée de la configuration (clavier, périphérique vocal, analogique, stylo-optique, etc ...).

Sous-rubrique : Caractérisation

On caractérise les entrées-utilisateur en fonction de la notion d'instruction telle qu'elle a été définie précédemment.

Pratiquement, une entrée peut être :

- une instruction de type d (instruction de production) ou c-branchement dans le logiciel.

- une «réponse», ce qui est désigné par "autre chose qu'une instruction ou une combinaison d'instructions" ; dans ce cas, il peut y avoir nécessité de construction personnelle de l'utilisateur, indépendamment des moyens logiciels.

- une combinaison d'instructions (examen d'une amorce de programmation).

Sous-rubrique : Organisation

On s'intéresse à la possibilité pour l'utilisateur d'enchaîner des entrées, et on tente d'apprécier par trois questions le degré d'intervention explicite du logiciel dans le discours-utilisateur.

Les deux premières questions cherchent à préciser s'il y a émission par le logiciel d'instructions suite à une entrée-utilisateur. Remarquons que lesdites instructions font partie de ce que nous appelons l'évaluation de l'entrée-utilisateur. Néanmoins, ici, il s'agit simplement de repérer la présence ou l'absence d'instructions données par le logiciel suite à une entrée de l'utilisateur, le problème général de l'évaluation étant traité par la

sous-rubrique suivante.

La dernière question de cette sous-rubrique concerne l'existence d'un mode différé.

Remarque :

L'examen des réponses aux rubriques "caractérisation des entrées", "organisation des entrées" et "validité des instructions" permet d'évaluer si l'utilisateur est maître des entrées et de leur ordre (dans le cadre d'un respect des règles).

Sous-rubrique : Evaluation

On se place du point de vue de l'utilisateur et on veut préciser ce qu'est l'évaluation de ses entrées.

Dans le cas où les entrées auraient été faites en mode différé, l'étude se situe à partir de l'exécution.

Le paragraphe "Effet de type 1" repère la production d'objets comme évaluation d'une entrée-utilisateur.

Le paragraphe "Effet de type 2" fait état de diverses données logicielles (c'est-à-dire instructions au sens défini pour cette recherche), qui peuvent apparaître.

Suite à la définition des divers types d'instructions, on peut dire que lorsqu'une entrée est une instruction, elle donne lieu à un effet de type 1 si elle est de type d (instruction de production), et à un effet de type 2 si elle est de type c (instruction de parcours-type branchement dans le logiciel).

Sous-rubrique : Statut des entrées-utilisateur ; degré de programmation laissé à l'utilisateur

Cette partie constitue une synthèse de questions précédentes et étudie la façon dont s'exprime la réponse au problème traité.

On dira qu'il y a présence de :

– "réponse à la machine", s'il existe des entrées-utilisateur qui ne sont pas des instructions (voir la rubrique de caractérisation des entrées).

– "commande de la machine de type pilotage", s'il est possible d'effectuer des entrées avec des instructions de type d (voir les rubriques de caractérisation des instructions et des entrées).

– "commande de la machine de type programme", s'il est possible d'enchaîner des entrées faites avec des instructions de type d (instructions de production) sans provoquer d'évaluation (voir les rubriques de caractérisation des instructions et des entrées).

L'analyse d'un logiciel selon cette sous-rubrique de la grille COM permet de préciser dans quelle mesure il est possible à un utilisateur d'exprimer un algorithme qu'il a construit.

Remarque :

Certains logiciels autorisent l'emploi d'instructions de parcours (type c) pour réaliser des entrées.

Ce genre d'entrées n'est pas retenu pour définir le mode commande.

En effet, serait alors conçu un mode commande qui porterait sur le parcours du logiciel. Il nous a semblé préférable de réserver le terme de «mode commande» pour ce qui concerne le contenu de l'activité. Le mode d'expression du parcours du logiciel, – qu'il serait d'ailleurs possible de modifier –, accentuera le cas échéant le caractère spécifique du logiciel. C'est en ce sens que l'on en tiendra compte.

3- ETUDE COMPARATIVE DES RESULTATS FOURNIS PAR L'ANALYSE A PRIORI DE LOGICIELS SELON LA GRILLE COM

Les pages suivantes décrivent l'élaboration de la typologie COM, en référence à l'analyse des neuf logiciels impliqués dans la mise au point de la typologie CONT.

Le travail s'est effectué dans le contexte déjà décrit en page 37 à propos de la typologie CONT.

Rappelons que le lecteur pourra consulter en annexe I un descriptif sommaire des logiciels, et en annexe III l'analyse de chaque logiciel.

Les éléments de réponses figurant dans les tableaux qui suivent proviennent de l'analyse rapportée en annexe III.

3a Etude comparative des résultats de la rubrique "Les instructions pour la réalisation de l'activité : les données du logiciel et de son environnement concernant la communication machine-utilisateur"

⇒ SOUS-RUBRIQUE "CARACTERISATION"

On peut établir le tableau suivant :

	Puce	Divis	Rect	Tache	Simul	Frise	Mos	Pictor	Logo
Instructions de définition de l'activité	X	X				(X)	X		
Instructions questions % au problème									
Instructions de parcours	X	X	X	X	X	X	X		
Instructions de production		X	X	X	X	X	X	X	X

Les deux premières lignes du tableau traitent d'instructions qui font référence à des problèmes déjà soulevés lors de l'étude des logiciels selon la grille CONT.

Les deux dernières lignes concernent des instructions qui paraissent déterminantes pour le classement des logiciels. Par conséquent, nous commencerons par l'analyse des deux dernières lignes du tableau ; l'examen des deux premières lignes apportera ensuite des précisions.

Les deux dernières lignes du tableau permettent de regrouper ainsi les logiciels :

Groupe I : PUCE

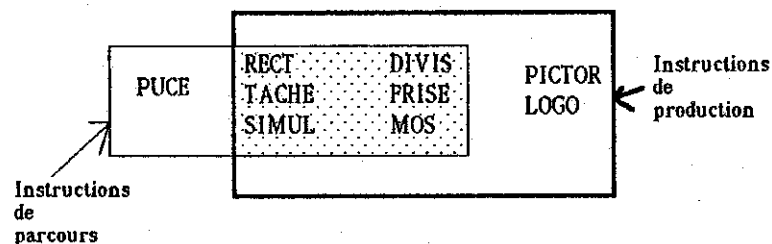
Groupe II : PICTOR - LOGO

Groupe III : DIVIS RECT TACHE SIMUL FRISE - MOS

Il est possible de trouver d'autres logiciels que PUCE, qui présentent les caractéristiques du groupe I. C'est pourquoi, nous exprimerons à ce stade la typologie recherchée en trois types :

- Type I:** logiciels - sans instructions de production
- avec instructions de parcours
- TYPE II:** logiciels - avec instructions de production
- sans instructions de parcours
- TYPE III:** logiciels - avec instructions de production
- avec instructions de parcours

Les deux premiers types sont opposés. Le dernier type est mixte :



Un type I' pourrait être proposé : logiciels sans instructions de production et sans instructions de parcours.

Si l'on considère le critère "instructions de production" comme déterminant, il constituerait un appauvrissement du type I. Notons qu'il présenterait au minimum une instruction du type "Donne ta réponse" et que d'autres considérations qui vont suivre autoriserait son assimilation au type I.

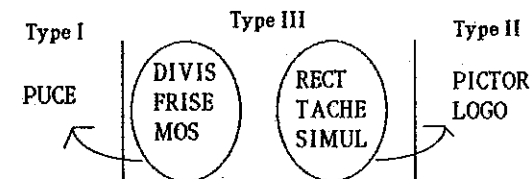
On peut aussi remarquer que les instructions de parcours des logiciels en cause ici sont de types branchement et repérage ; il existe des logiciels, essentiellement de type I, qui ne font référence qu'au type repérage.

L'examen des deux premières lignes du tableau permet de compléter ce qui précède :

Les logiciels ne présentent pas d'instructions "questions à l'utilisateur par rapport au problème traité". Il serait possible qu'il en soit autrement, essentiellement en type I, ces instructions correspondant aux "questions conceptuelles" évoquées lors de l'analyse selon la grille CONT.

Le type I admet des instructions de définition de l'activité à l'inverse du type II.

Le type III se rattache au type I ou II pour l'existence d'instructions de définition de l'activité :



Les flèches sur ce schéma indiquent la tendance des logiciels vers un type.

Nous avons déjà observé lors du travail concernant la rubrique "donnée des consignes" de l'analyse CONT que le comportement des logiciels du type III pourrait être modifié en ce qui concerne les consignes de définition de l'activité : on peut en inclure dans un logiciel qui n'en présente pas, ou au contraire les supprimer d'un logiciel qui en admet, et laisser l'enseignant les émettre.

• MISE AU POINT

Nous préciserons la typologie COM de la façon suivante :

Type I : logiciels à plusieurs genres d'instructions, à l'exclusion du genre "instruction de production".

Les genres "instructions de définition de l'activité" et à des degrés divers "instructions de parcours" sont toujours présents.

Le genre "instructions questions par rapport au problème traité" existe le cas échéant.

Type II : logiciels à 1 seul genre d'instructions : "instructions de production".

Type III : logiciels à plusieurs genres d'instructions.

Les genres "instructions de production" et "instructions de parcours" sont toujours présents.

Les autres genres existent le cas échéant.

• REMARQUES ET QUESTIONS

En types I et III coexistent plusieurs genres d'instructions. Comment l'utilisateur va-t-il gérer ce fait ?

Comme il en a déjà été question lors de l'explicitation des rubriques de COM, suivant son genre une instruction sera donnée par la machine à l'utilisateur, ou sera donnée par l'utilisateur à la machine. En type II, il n'y a

que des instructions de production ; par conséquent, en type II, l'utilisateur donne des instructions à la machine. En type I et III, il y a des instructions de définition de l'activité ou de parcours du logiciel ; par conséquent, en type I et III, l'utilisateur reçoit des instructions de la machine. En outre, en type III, il y a des instructions de production ; par conséquent, en type III, l'utilisateur donne aussi des instructions à la machine.

Apparaissent deux possibilités pour l'utilisateur : donner des instructions ou en recevoir. Comment interviennent-elles, s'organisent-elles dans les divers types ? A quoi peuvent-elles être employées ? Quelles attitudes engendrent-elles ? Quels sont les enjeux ?

La partie II de la recherche tentera d'éclairer ces divers points.

⇒ SOUS-RUBRIQUE "SUPPORT DES INSTRUCTIONS"

	Puce	Divis	Rect	Tache	Simul	Frise	Mos	Pictor	Logo
Support écran	X	X	X	X	X	X	X	X	
Est-il nécessaire ?		X				X		X	
Support document		(X)	X	X	X	(X)		X	X

Les logiciels DIVIS et FRISE n'étaient pas commercialisés à la date de leur analyse et ne disposaient donc pas encore de document d'accompagnement, d'où les réponses entre parenthèses.

On note que :

– Les logiciels de type I (PUCE, comme d'autres non étudiés ici) présentent les instructions à l'écran.

– Les logiciels de type II présentent les instructions sur un document.

Remarquons que pour PICTOR, l'usage du crayon optique rend nécessaire l'emploi de l'écran pour permettre l'"écriture".

– Les logiciels de type III peuvent utiliser les deux genres de supports. FRISE et DIVIS semblent plus près du type II, MOS du type I.

Pour RECT, TACHE, SIMUL, ce ne sont pas les mêmes instructions qui

sont sur l'écran et sur le document. Ceci semble confirmer l'ambivalence du type III, ici au niveau du support des instructions.

Une question se pose :

Un genre d'instruction rend-il un genre de support nécessaire ? Y-a-t-il des logiciels pour lesquels on pourrait aisément remplacer un support par un autre ?

Intrinsèquement, les instructions qui peuvent être des entrées utilisateur (production et parcours-branchement) ne nécessitent pas le support écran (sauf, en cas de recours obligé à un périphérique d'entrée adressant l'écran : souris, crayon optique ...).

En conséquence, nous répondrons à la question posée ci-dessus par une spécification des divers types.

• MISE AU POINT

En type I : la présence d'instructions à l'écran est rendue obligatoire par l'absence d'instructions de production.

En type II : l'absence d'instructions à l'écran et l'existence d'undocument est caractéristique du type : il n'existe qu'un seul genre d'instructions, "instructions de production" ; celles-ci peuvent difficilement être mises à l'écran dans la plupart des cas compte-tenu de leur nombre, leur forme, leur complexité.

Lorsqu'elles le sont, c'est généralement en raison du périphérique d'entrée qui rend ce fait nécessaire. Dans ce cas, elles peuvent être peu nombreuses ou organisées en divers sous-ensembles auxquels on a accès du fait même de la désignation de l'une des instructions

PICTOR propose ainsi 3 lignes d'instructions.

En type III : Les instructions de production peuvent se trouver sur un document.

C'est le cas pour SIMUL et RECT.

FRISE et DIVIS présentent leurs instructions à l'écran, mais le crayon optique rend ce fait obligatoire.

Les autres instructions peuvent se trouver sur l'un ou l'autre support, plus facilement sur document si le logiciel présente un seul type d'activité que s'il en présente un ensemble.




On pourrait le cas échéant modifier un logiciel quant au support emprunté par ces dernières instructions ; le logiciel se caractériserait alors plus nettement en type I ou II.

• **QUESTION**

Comment l'utilisateur prend-il en compte la ou les formes de support qui lui sont imposées ?

⇒ **SOUS-RUBRIQUE "VALIDITE DES INSTRUCTIONS"**

Comme il a déjà été signalé à l'explicitation de cette rubrique, cette question se pose pour les instructions qui peuvent être des entrées utilisateur, c'est-à-dire de type d (instructions de production), et de type c (instructions de parcours-branchement).

	Type I	Type III						Type II	
	Puce	Divis	Rect	Tache	Simul	Frise	Mos	Pictor	Logo
Toutes sont valides sur tout le logiciel								X d	X d
Certaines sont valides sur une partie du logiciel		X d	X d	X d,c	X d	X d,c	X d		
Certaines sont valides à un seul endroit du logiciel		X c	X d,c		X d,c				

d et c précisent le genre d'instructions (production ou parcours branchement)

Cette sous-rubrique précise la validité des instructions en relation avec les entrées-utilisateur. Corrélativement, elle permet de conclure dans certains cas sur la nature des entrées utilisateur (cette étude étant poursuivie au cours des autres rubriques).

L'ensemble des renseignements ici obtenus fournit les premiers éléments d'une définition des logiciels en terme de structure, du point de vue de la communication proposée à l'utilisateur avec la machine. Une telle définition en terme de structure se veut un moyen d'approche de la position de l'utilisateur avant l'utilisation effective. Pour exprimer dorénavant les structures des logiciels telles que nous les dégagerons, nous recourrons à une schématisation dont le principe est donné en annexe V et dans le fascicule d'accompagnement.

Avec le critère de validité des instructions nous retrouvons les trois types dégagés auparavant (ceci cautionne les critères retenus pour amorcer la

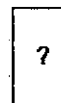
typologie). L'examen du tableau permet de les spécifier.

• **MISE AU POINT**

En type I, les instructions ne peuvent être des entrées-utilisateur. L'utilisateur ne dispose d'aucun corpus préalable pour communiquer avec la machine. Les entrées-utilisateur ne sont pas des instructions. Que sont-elles ?

On va schématiser ainsi la structure du type I :

(Cas de PUCE)



Avant l'utilisation effective, l'utilisateur ne dispose de rien.

En type II, toutes les instructions qui peuvent être des entrées utilisateur sont valides sur tout le logiciel. L'utilisateur dispose d'un corpus d'instruction pour communiquer avec la machine. Les entrées utilisateur reçoivent toutes des instructions.

On va schématiser ainsi la structure du type II :

(Cas de PICTOR, LOGO)



Pour ce type, l'utilisateur peut avoir connaissance des diverses instructions disponibles avant de débiter son activité. Nous avons d'ailleurs constaté lors de l'examen de la précédente sous-rubrique qu'elles se trouvaient toujours au moins sur un document accompagnant le logiciel. Par conséquent, avant l'utilisation effective, l'utilisateur dispose de tous les renseignements possibles pour communiquer et il sait que ceux-ci sont valides pour toute son activité.

En type III, il existe au moins un groupe d'instructions valides sur une partie du logiciel.

Il existe des entrées qui reçoivent des instructions, mais toutes ne sont pas valides sur tout le logiciel, comme en type II.

Par conséquent, il existe des entrées non couvertes par des instructions ou (inclusif) des entrées couvertes par des groupes d'instructions différents.

Ceci conduit à poser les problèmes suivants :

– La totalité des instructions couvre-t-elle la totalité des entrées ? Que reçoivent les entrées non couvertes par des instructions ? Quelle est

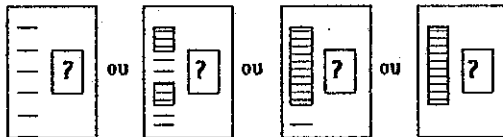
l'importance numérique des entrées non couvertes par des instructions ?

– Combien y-a-t-il de groupes d'instructions valides sur combien de parties du logiciel ? Comment les groupes d'instructions se distribuent-ils sur les entrées ?

La structure d'un logiciel de type III sera plus ou moins proche de celle du type II suivant les réponses à ces questions. On pourra avoir localement une structure type II.

L'examen de la 3e ligne du tableau précise s'il existe des instructions valides à un seul endroit du logiciel (structure éloignée de celle du type II).

Ce qui précède permet de proposer pour le type III, des schématisations de structures de la façon suivante :



Remarques

Rappelons que le graphisme avec un point d'interrogation désigne des entrées non couvertes par des instructions. Il peut être absent de la schématisation de la structure des logiciels de type III.

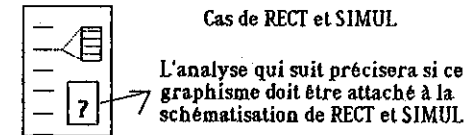
Les groupes d'instructions affectables aux divers endroits schématisés par un trait sont différents.

Les groupes d'instructions affectables aux diverses parties rayées sont différents ou identiques.

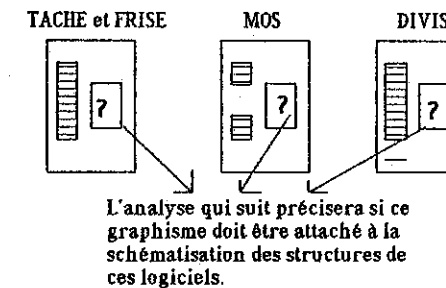
Suivant la façon dont les instructions sont mises à disposition de l'utilisateur (voir la sous-rubrique "Support des instructions"), et suivant la proximité de la structure du logiciel à une structure globale ou locale de type II, avant l'utilisation effective ou après un certain temps d'emploi, l'utilisateur disposera de renseignements pour communiquer sur certaines parties du logiciel ; se posera alors la question de la gestion d'ensemble de cette communication.

Pour les logiciels analysés, il est possible de préciser :

– on peut regrouper RECT et SIMUL sous le caractère dominant de l'existence de certaines instructions valides à un seul endroit du logiciel :



– on peut regrouper FRISE, MOSAIQUE, TACHE, DIVIS sous le caractère dominant de l'existence de certaines instructions valides sur une partie du logiciel :



Signalons pour DIVIS, le fait que l'entrée marquée d'un trait reçoit une instruction faisant partie d'un groupe d'instructions de parcours : la structure est donc encore très proche de celle de TACHE.

A ce stade de l'étude, TACHE, FRISE, DIVIS sont donc très proches du type II du point de vue de la structure. MOS semble l'être plus localement.

• **QUESTIONS**

– S'il existe des entrées non couvertes par des instructions, quereçoivent-elles ? Peuvent-elles recevoir des objets différents ?

– Que permettent les entrées faites avec des instructions ? Chacune de ces entrées se fait avec une instruction choisie dans un groupe d'instructions. Combien y-a-t-il d'instructions dans un groupe affectable à un endroit ou sur une partie d'un logiciel ?

La réponse à ces questions permettrait de distinguer les logiciels de type II, comme les logiciels de type III, sur des critères de puissance du corpus mis à la disposition de l'utilisateur pour communiquer avec la machine.

– Existe-t-il autre chose dans le logiciel que des possibilités d'entrée utilisateur ? Comment cela intervient-il ? Comment l'utilisateur gère-t-il cet autre champ, le cas échéant en liaison avec le précédent ?

– Comment l'utilisateur perçoit-il et gère-t-il les diverses structures exposées dans cette partie ?

– Comment et quand l'utilisateur apprend il à utiliser le corpus d'instructions qui est mis à sa disposition en types II et III ? L'apprentissage est-il nécessaire ? Comment mesurer son importance suivant les logiciels ? Se réduit-il à celui d'instructions ? Comment se situe l'activité mathématique engagée en utilisant un logiciel par rapport à cet apprentissage ?

3b Etude comparative des résultats de la rubrique "Les entrées-utilisateur"

⇒ SOUS-RUBRIQUE "CARACTERISATION"

	Type I		Type III					Type II	
	Puce	Divis	Rect	Tache	Simul	Frise	Mos	Pictor	Logo
Une entrée peut-elle être une instruction ?		X	X	X	X	X	X	X	X
Une combinaison d'instructions ?								(X)	X
Autre chose ?	X		X	X			(X)		

Rappels concernant les parenthèses dans le tableau :

– Pour Pictor, une entrée n'est pas à strictement parler une combinaison d'instructions ; néanmoins pour obtenir certains objets, il faut effectuer plusieurs entrées successives. En ce sens, la production peut être comprise comme réalisée avec une combinaison de plusieurs instructions.

– Pour MOSAÏQUE, l'utilisateur effectue certains branchements (accès par exemple à une modification de sa réalisation) en répondant par oui ou par non. Ces réponses ne concernent pas directement le contenu de l'activité et peuvent donc être estimées secondaires pour la caractérisation. En outre, il est possible de les assimiler à des instructions de branchement (parcours du logiciel).

L'examen du tableau, permet de poursuivre la spécification des trois types.

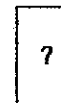
• MISE AU POINT

Type I : Une entrée n'est pas une instruction.

Ce type exige une construction personnelle, de l'utilisateur n'employant pas de données brutes du logiciel ou de son environnement.

Toutes les entrées ne sont pas nécessairement des constructions personnelles de l'utilisateur : il peut s'agir d'expression de choix relatifs à des branchements. Néanmoins, il ne peut en être ainsi sur tout le logiciel.

La schématisation de la structure du type I reste :



Type II : Une entrée est une instruction ou une combinaison d'instructions.

Il s'agit ici d'une utilisation des données du logiciel ou de son environnement.

Notons que si la combinaison d'instructions est possible, l'anticipation est une possibilité attachée au logiciel.

La schématisation de la structure du type II reste :



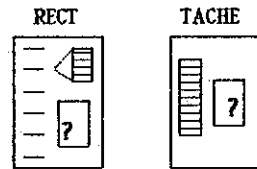
Type III : Une entrée utilisateur peut être une instruction, mais toutes ne le sont pas.

Nous définirons deux sous-types :

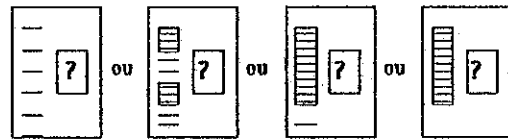
Sous-type III-I : Une entrée peut être une instruction ou autre chose.

On note la proximité de ce sous-type au type I, comme la caractéristique d'entrées non réalisées à l'aide d'instructions qu'il emprunte au type II.

Pour les logiciels mis en cause, on a les structures suivantes :

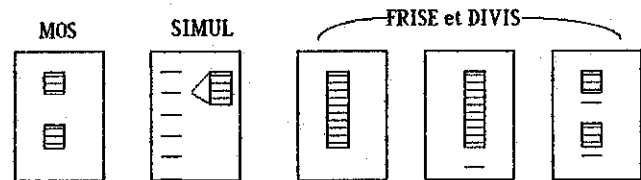


D'une façon générale, le sous-type III-I se caractérise par une schématisation de sa structure de genre mixte : certaines entrées reçoivent des instructions, d'autres non.



La proximité au type II sera plus ou moins grande suivant l'importance des entrées admettant des instructions.

Sous-type III.II : Une entrée est une instruction. Ce sous-type est très proche du type II. Sa structure peut être comprise comme un "éclatement" de celle relative au type II :



(Suivant que l'on examine un problème ou un cursus de problèmes)

• QUESTIONS

— Pour les logiciels du sous-type III-I, l'utilisateur effectue des entrées de deux sortes. Comment gèrera-t-il la double demande qui lui est faite ? Comment se situera-t-il ?

— On posera en fin d'étude des structures des logiciels, une question générale sur les enjeux des divers types.

⇒ SOUS-RUBRIQUE "ORGANISATION"

	Type I	Type III					Type II		
		III-I		III-II					
Peut-on enchaîner des entrées	Puce	Rect	Tache	Divis	Simul	Frise	Mos	Pictor	Logo
Sans recevoir d'instructions autres qu'une invite ?						X		X	X
Sans recevoir d'instructions autres que la précédente ?		(X)	X	X	(X)		(X)		
Sans provoquer d'évaluation ?								(X)	X

Rappels concernant les parenthèses dans le tableau :

— Pour PICTOR, l'obtention de certains objets (un rectangle par exemple) nécessite un enchaînement de plusieurs instructions ; l'évaluation n'intervient qu'à la dernière.

— Pour RECT et SIMUL, la réponse est "oui" mais sur une partie optionnelle du logiciel. Le caractère dominant implique une réponse négative.

— Pour MOS, il peut s'agir de la même instruction appliquée à des objets différents ; elle peut être entendue comme identique dans une certaine mesure.

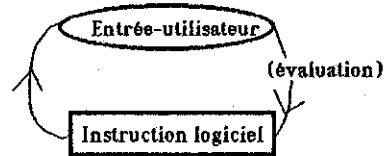
Les deux premières questions interrogent sur l'émission par la machine d'instructions suite à une entrée utilisateur.

Le problème général de l'évaluation de l'entrée-utilisateur sera vu au cours de l'examen de la rubrique "évaluation" : ici, il s'agit simplement de repérer la présence d'instructions émises par le logiciel dans le discours machine-utilisateur. Néanmoins l'évaluation est d'ores et déjà signalée comme élément fondamental ; lors de l'analyse des effets de l'évaluation, son contenu sera intégré aux schémas que nous allons maintenant mettre en place. Les instructions dont il sera ici question peuvent faire partie de l'expression de l'évaluation.

La dernière question s'intéresse à l'existence d'un mode différé.

• MISE AU POINT

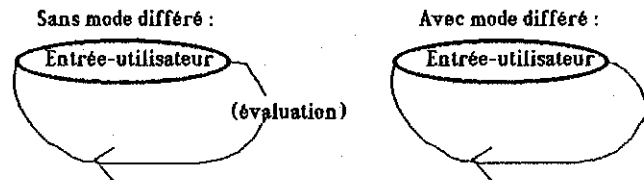
Le type I : Après une entrée, l'utilisateur reçoit une instruction.
La communication machine-utilisateur peut se résumer ainsi :



Ceci est vrai sur tout le logiciel et cohérent avec la structure dégagée auparavant.

Le type II : Après une entrée, l'utilisateur ne reçoit pas d'instructions autres qu'un éventuel symbole d'invite, — sauf un message d'erreur en cas d'incorrection syntaxique/sémantique par rapport au discours.

La communication machine-utilisateur peut se résumer ainsi :

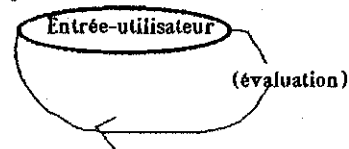


Ce qui précède est vrai sur l'ensemble du logiciel, et est cohérent avec la structure dégagée auparavant.

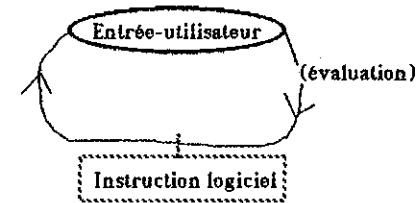
LOGO présente cette structure, de même que PICTOR de façon plus élémentaire et ponctuelle.

Le type III : On ne retrouve pas clairement les deux sous-types III-I et III-II proposés plus haut.

— Il existe des logiciels très proches du type II, mais qui n'offrent pas de mode différé (cas de FRISE).

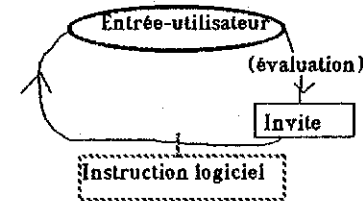


— Le schéma qui précède peut ne caractériser qu'une partie du logiciel. Certaines entrées peuvent donner lieu à l'émission d'une instruction logiciel (Cas de FRISE).

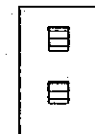


Pour FRISE par exemple, nous avons vu lors de l'étude de la sous-rubrique "Caractérisation des entrées utilisateur" que des instructions de parcours-repérage étaient affichées à certaines phases.

— La structure de certains logiciels répond également à la schématisation ci-dessus, si l'on assimile l'émission d'une même instruction à celle d'un symbole d'invite (Cas de DIVIS, TACHE, MOS).

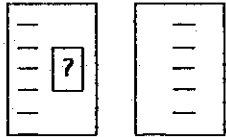


Remarquons que les logiciels du type III que nous venons d'évoquer admettent tous localement au moins une structure apparentée au type II :

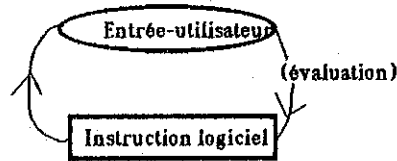


— Certains logiciels répondent localement au schéma comportant invite et instruction du logiciel, mais sont globalement apparentés au type I (Cas de RECT et SIMUL).

Ce sont ceux dont la structure peut se décrire :



Un schéma type I leur est associé :



En conclusion, pour le type III, on constate que, en conformité avec les résultats de l'étude portant sur la "caractérisation des entrées utilisateur", les distinctions pour l'organisation des entrées utilisateur relèvent de la structure mise à jour en examinant la validité des instructions. Les logiciels s'apparentent au type I ou II plus ou moins localement.

Nous ferons l'hypothèse, qu'une modification visant à leur donner une structure de type II est possible : il suffirait, en s'intéressant aux groupes d'instructions valides, de définir les règles de leur emploi (en particulier, règles concernant l'ordre) ; on pourrait les inscrire par exemple sur un document d'accompagnement.

Les deux sous-types III-I et III-II seront maintenus : en effet, sous l'hypothèse qui précède, les logiciels III-II pourraient (idéalement) être transformés en logiciels II complètement ; les logiciels III-I ne pourraient l'être que localement en raison de l'existence d'entrées ne recevant pas d'instructions.

• ***SYNTHÈSE***

— Pour exprimer la structure d'un logiciel du point de vue des formes de la communication de l'utilisateur avec la machine, nous établissons deux schémas :

- un schéma caractérisant les entrées, et la validité des instructions.
- un schéma caractérisant l'organisation des entrées.

Pour schématiser véritablement les structures des trois types, il est nécessaire de compléter le deuxième schéma d'informations concernant

l'évaluation. Cette étude sera faite par la sous-rubrique suivante. C'est pourquoi, nous différons l'établissement des schémas en question.

— Il est néanmoins possible d'effectuer une synthèse partielle et provisoire concernant la position de l'utilisateur :

En type I, l'utilisateur ne dispose de rien et ne décide de rien.

En type II, l'utilisateur dispose d'instructions valides sur l'ensemble du logiciel. La structure dégagée à ce stade du travail permet de penser qu'il décide du moment d'une entrée et de l'ordre dans lequel il va effectuer ses diverses entrées (sous respect des règles d'utilisation des instructions). En outre, il peut employer un mode différé.

En type III, l'utilisateur a localement les mêmes possibilités qu'en type II (à l'exclusion du mode différé) avec des groupes d'instructions associées à des parties du logiciel plus ou moins proches de la totalité, éventuellement réduites à une entrée. Il peut localement ne disposer de rien pour effectuer une entrée (entrée non couverte par une instruction).

• ***QUESTIONS***

Comment intervient l'évaluation ?

Quelle est la tâche de l'utilisateur dans une situation qui emploie un logiciel d'un certain type ?

Quelle est l'influence de la structure sous-jacente ?

⇒ SOUS-RUBRIQUE "EVALUATION"

Dans cette partie, nous allons examiner en quoi consiste l'évaluation signalée sur les schémas concernant l'organisation des entrées utilisateur.

Certains aspects ont déjà été abordés lors de l'étude de la validation de l'activité par rapport au contenu par la grille CONT.

	Type I	Type III						Type II	
	Puce	III-I			III-II			Pictor	Logo
		Rect	Tache	Divis	Simul	Frise	Mos		
Evaluation % au discours de l'utilisateur	X	X	X	X	X	X	X	X	X
Evaluation % à l'attendu conceptuel	X		X	X					
Effet 1 : production		X	X	X	X	X	X	X	X
Effet 2 : donnée du logiciel	X	X	X	X	X	X	X		
Message % à l'attendu conceptuel	X		X	X					
Score			X	X					
Temps		X							
Outil spécifique	X		(X)						
Visualisation	X								
Autres instructions	X	X	X	X	X	X	X		
Nombre de types d'effet	1	2	2	2	2	2	2	1	1
Support graphique	X	X	X	X	X	X	X	X	X
Support sonore	X			X					
Support mobile									

Sous-rubrique : "Caractérisation"

- Il y a toujours évaluation par rapport au discours utilisateur.

En cas d'erreur, un message est envoyé.

Notons que l'existence de cette évaluation ne prouve en rien sa pertinence pour l'activité. La deuxième partie de la recherche s'intéresse à ce problème.

- Certains logiciels effectuent une évaluation par rapport à l'attendu conceptuel.

Il s'agit des logiciels de type I (cas de PUCE) et de type III-I (TACHE à strictement parler et RECT s'il comportait une analyse de réponses ; RECT de type III-I possède ainsi une caractéristique en "moins" et s'affilie ici au type II). DIVIS appartient à cette catégorie, néanmoins l'évaluation par rapport à l'attendu conceptuel pourrait être supprimée : DIVIS de type III-II possède ainsi une caractéristique en "plus", de type I.

En résumé, les logiciels de types I et III-I effectuent une évaluation par rapport au discours utilisateur et par rapport à l'attendu conceptuel.

Les logiciels de type II et III-II effectuent une évaluation seulement par rapport au discours utilisateur.

En outre, on peut noter à nouveau l'intervention de critères de type I ou de type II dans un logiciel de type III.

Sous-rubrique : "Effet"

Cette partie apporte des compléments d'information sur la structure des logiciels des divers types, et corrélativement sur la position de l'utilisateur.

Elle souligne l'expression de l'évaluation suite à une entrée de l'utilisateur.

On rappelle que les instructions du logiciel notées sur les schémas de caractérisation des entrées-utilisateur peuvent faire partie de l'expression de l'évaluation : il en sera tenu compte dans la schématisation que nous proposons.

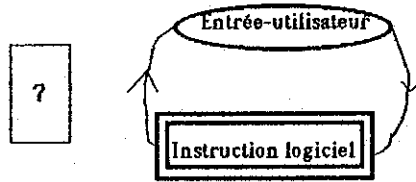
La lecture des lignes "nombre de types d'effet", "Effet 1 : production", "Effet 2 : donnée du logiciel" réaffirme les trois types de logiciels mis à jour antérieurement.

• **MISE AU POINT**

Le type I : Logiciels à un seul type d'effet : envoi d'une donnée du logiciel, c'est-à-dire d'une instruction. L'instruction émise peut avoir divers contenus qu'il est possible de préciser (voir les six lignes de la grille d'analyse qui suivent la ligne "Effet 2").
Ainsi, l'évaluation d'une entrée faite avec autre chose qu'une instruction consiste en une instruction.

La schématisation de la structure des logiciels de type I est la suivante :

(Cas de PUCE)

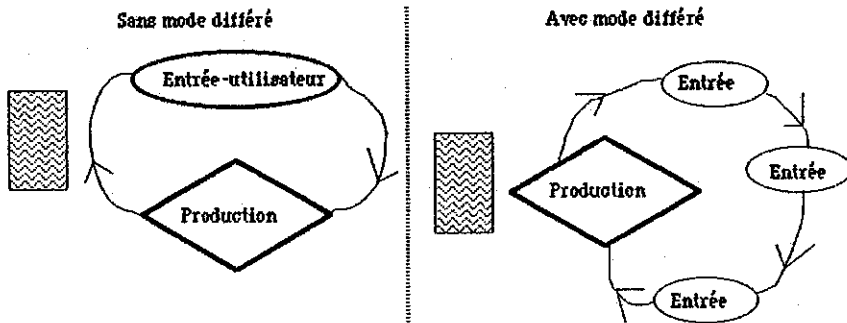


L'instruction machine peut être un message par rapport à l'attendu conceptuel : cela est signalé par le double cadrage.

Le type II : Logiciels à un seul type d'effet : production d'objets commandés par l'entrée utilisateur. Cas par cas, il est possible de préciser ces objets (la partie II de la recherche y reviendra).
Ainsi, l'évaluation d'une entrée faite avec une instruction de production consiste en une production.

La schématisation de la structure du type II est :

(Cas de PICTOR, LOGO)



Le type III : Logiciels à deux types d'effet : production d'objets commandés par l'utilisateur et envoi d'une donnée du logiciel (instruction).

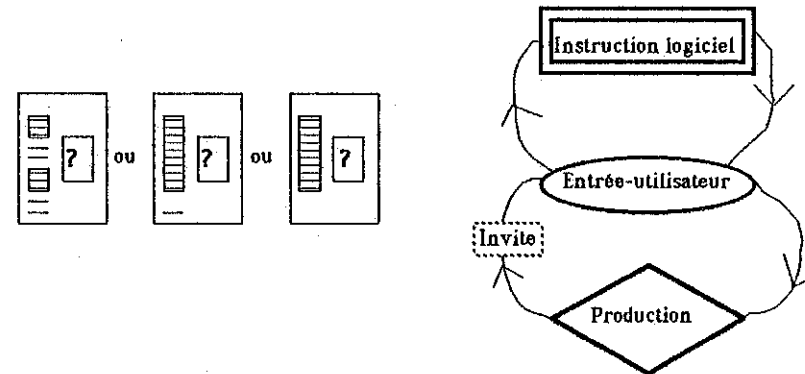
Ainsi, en se référant au type III.II (essentiellement SIMUL, FRISE, MOS) et à ce qui précède relativement à l'évaluation des entrées, on note que l'évaluation d'une entrée faite avec une instruction de parcours consiste en une instruction, tandis que l'évaluation d'une entrée faite avec une instruction de production consiste en une production.

De ce fait, il est intéressant de signaler ici que les instructions "message par rapport à l'attendu conceptuel" et "score" de DIVIS interviennent suite à une entrée faite avec une instruction de production, et suite à la production en cause. Cela confirme le caractère inutile intrinsèquement de ces deux données dans le logiciel et pose la question de leurs rôles.

La séparation du type III en deux sous-types s'appuie sur les questions déjà traitées ; elle est confirmée par les six lignes suivant la ligne "Effet 2" :

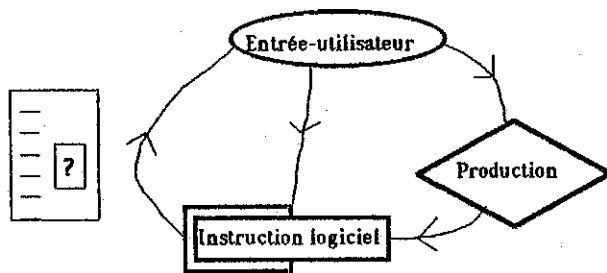
- La schématisation du sous-type III-I est :

(Cas de TACHE)



L'instruction machine peut être un message par rapport à l'attendu conceptuel.

Ou encore :



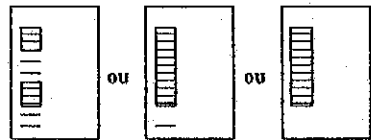
(Cas de RECT)

Pour RECT, l'instruction machine est une instruction de repérage ; néanmoins, elle pourrait consister en un message par rapport à l'attendu conceptuel.

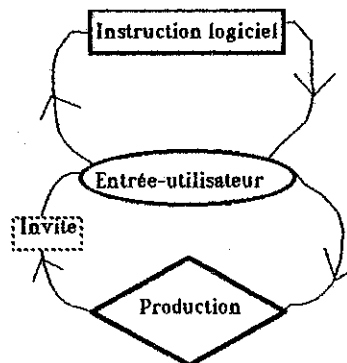
La constante en sous-type III-I est l'existence d'un cycle strictement de type I.

Toutefois, il peut exister un cycle de type II. Celui-ci peut être strict ou mixte, c'est-à-dire qu'il admettra production et instruction du logiciel (autre qu'un message par rapport à l'attendu conceptuel).

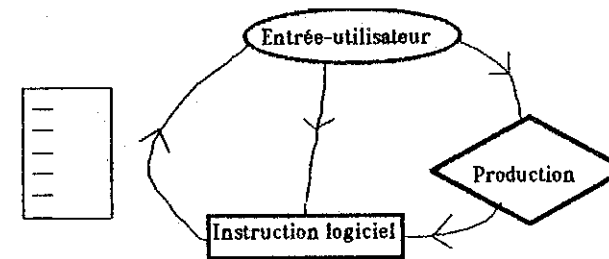
- La schématisation du sous-type III-II est proche de celle du sous-type III-I :



(Cas de FRISE, MOS et DIVIS)



- ou encore :



(Cas de SIMUL)

Notons que dans les deux cas, l'instruction du logiciel n'est pas un message par rapport à l'attendu conceptuel : cela est signalé par l'absence de double cadrage.

Il s'agit d'une caractéristique constante : en sous-type III-II, il n'y a pas de cycle strict de type I.

Des cycles stricts ou mixtes (c'est-à-dire comprenant une instruction du logiciel) de type II sont présents.

L'existence de cycles stricts I ou II peut être un élément pour juger de la proximité d'un logiciel de type III au type I ou II.

Le nombre de cycles I ou II stricts ou mixtes et leur organisation peut être apprécié en effectuant des schémas pointus : on pourrait alors mesurer plus finement la proximité aux types de référence.

Par exemple, avec "FRISE" le nombre de cycles II s'avère très nettement supérieur à celui des cycles I.

Sous-rubrique : "Support"

Tous les logiciels ici en cause émettent l'évaluation sur support graphique (écran).

Certains logiciels emploient pour exprimer l'évaluation d'entrées spécifiques (réponses au problème traité en particulier) un double support graphique et sonore.

• QUESTIONS

Nous poserons ici des questions concernant l'ensemble de la sous-rubrique "Evaluation".

- Que vont permettre les divers genres de cycles proposés à l'utilisateur ?

- Comment celui-ci gèrera-t-il un logiciel à plusieurs types de cycles ?

⇒ SOUS-RUBRIQUE "STATUT DES ENTREES-UTILISATEUR ; DEGRE DE PROGRAMMATION LAISSE A L'UTILISATEUR"

	Type I	Type III						Type II	
	Puce	III-I		III-II				Pictor	Logo
		Rect	Tache	Divis	Simul	Frise	Mos		
Y-a-t-il réponse à la machine ?	X	X	X				(X)		
Commande de type pilotage ?			X	X	X	X	X	X	X
Commande de type programme ?								(X)	X
Réponse au problème : en mode réponse	X	X	X						
Réponse au problème : en mode commande				X	X	X	X	X	X

A propos des parenthèses présentes dans le tableau :

- Pour MOSAÏQUE, les entrées-utilisateur ayant statut de réponse sont essentiellement des branchements ; elles pourraient être conçues différemment, et avoir qualités de commande explicite.
- Pour PICTOR, l'obtention de certains objets nécessite plusieurs entrées successives. Il ne s'agit pas à strictement parler de commande de type programme, mais il est possible que l'utilisateur l'apprecie localement comme tel. Toutefois, PICTOR ne permettant pas la création d'objets logiciels (mais de fichiers de données), nous considérerons la réponse à la question comme globalement négative.

L'examen des deux premières lignes du tableau permet de dire que :

- Les logiciels de type I ne permettent pas une commande de la machine de type pilotage : il y a réponse à la machine.

- Les logiciels de types II et III permettent la commande de la machine de type pilotage.

La question du statut des entrées utilisateur est fondamentale pour la mise au point de la typologie.

En effet, les trois derniers critères ("y-a-t-il présence de commande de la machine de type programme ?" "L'expression de la réponse au problème traité se fait-elle en mode réponse ?" "L'expression de la réponse au problème traité se fait-elle en mode commande ?"), sont tout spécialement déterminants : l'examen des trois derniers critères va autoriser un remodelage de la typologie, que l'on peut envisager comme une conclusion, à laquelle s'intègre l'ensemble de l'analyse.

La nouvelle forme de la typologie devrait permettre de classer tout nouveau logiciel sur la base d'une analyse un peu structurée des résultats de son traitement par COM.

Nous allons tout d'abord expliciter le remodelage de la typologie par l'examen des réponses aux trois derniers critères.

• MISE AU POINT

LE TYPE I

Il s'agit de logiciels «réponse».

La réponse au problème traité s'effectue en mode réponse.

On peut distinguer :

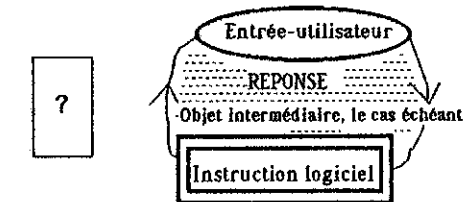
Le sous-type I-I

Toutes les entrées sont en mode réponse.

S'il y a activité sur un objet intermédiaire (c'est-à-dire, un objet qui n'est pas celui du problème initialement posé), elle se déroule en mode réponse.

Schématisation :

(Cas de PUCE)



Remarque

Pour PUCE, il y a possibilité de parcours ramifié du logiciel. Les choix sont effectués en mode réponse.

Il peut exister des logiciels avec parcours ramifié, où les choix concernant le parcours se font avec un statut de commande avec des instructions de parcours (type c).

Il peut aussi exister des logiciels sans possibilité de parcours ramifié.

En sous-type I-I, il y a introduction de données par l'utilisateur dans un algorithme ; il n'y a pas de possibilité d'exprimer un algorithme. Le degré de programmation laissé à l'utilisateur est nul.

Le sous-type I.II

Nous choisissons délibérément de renommer « sous-type I.II », l'ancien sous-type III-I.

Ce changement de notation, qui n'affecte pas les précédentes observations concernant les logiciels de ce sous-type, nous paraît nécessaire. En effet, une telle notation I-II souligne l'existence de points communs avec le type II. Elle consacre également l'affiliation au type I : la réponse au problème traité est communiquée à l'ordinateur en mode réponse. Souhaitant privilégier le mode d'expression de la réponse au problème traité comme élément fondamental par rapport à l'activité, nous avons donc opté pour cette transformation de dénomination.

Pour ce sous-type, il existe une possibilité de commande de la machine de type pilotage : il ne s'agit pas seulement du parcours du logiciel, mais conformément à la définition du mode commande, ici, des entrées peuvent être faites avec des instructions de production (type d).

En conséquence, les logiciels de sous-type I-II permettent, voire imposent la construction d'objets intermédiaires en mode commande.

Précisons :

En ce qui concerne RECT et TACHE, il y a exigence sur au moins une partie du logiciel : le sous-type I-II est caractérisé essentiellement sur cette base.

C'est pourquoi, on parlera d'appartenance à un sous-type I-II⁺, lorsqu'il y a exigence sur au moins une partie du logiciel. Remarquons que pour ce sous-type I-II⁺, les cycles dominants semblent être les cycles « production objet intermédiaire », dont on s'échappe vers un cycle « instruction-réponse » pour exprimer la réponse au problème traité.

En tout état de cause, en sous-type I-II⁺, les deux genres de cycles (production, instruction-réponse) sont nécessairement employés.

On parlera d'appartenance à un sous-type I-II⁻, lorsque la réalisation d'un objet intermédiaire en mode commande est optionnelle.

Ici, on est plus proche du sous-type I-I. Les cycles dominants semblent être les cycles « instruction-réponse », dont on s'échappe, si on le souhaite, vers des cycles « production objet intermédiaire ».

En sous-type I-II⁻, les deux genres de cycles ne sont pas nécessairement employés.

PUCE se classerait en I-II eu égard à ses options (calculatrice, puce écran). Remarquons néanmoins que celles-ci n'offrent que peu de possibilités : l'option calculatrice produit un objet intermédiaire sur un seul cycle ; l'option puce écran est encore plus pauvre.

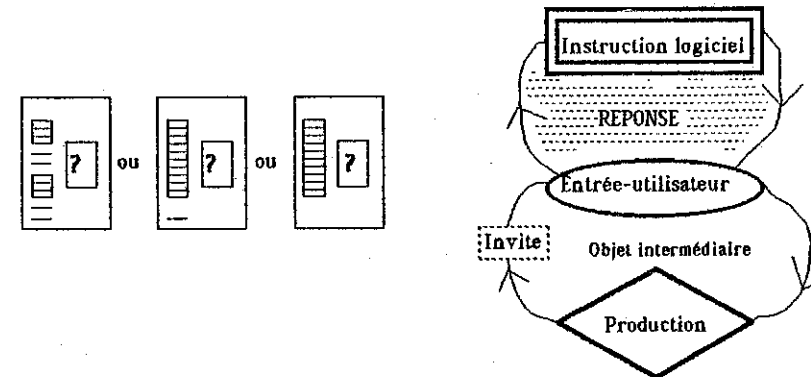
Il s'agirait d'un logiciel I-II très limité. Ceci peut expliquer les résultats d'analyse que nous proposons et son classement en sous-type I-I.

Avec les sous-types I-II, il y a dans une plus ou moins grande mesure (en relation aussi avec les instructions de production disponibles et le nombre d'entrées autorisées), la possibilité d'exprimer un algorithme et d'anticiper pour construire un objet intermédiaire.

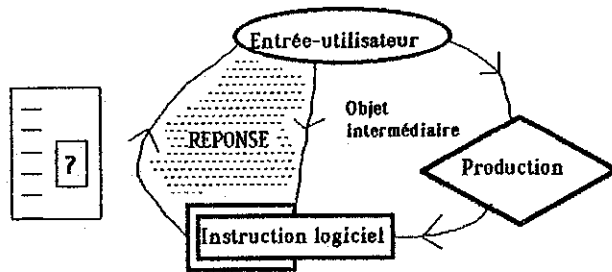
On peut parler d'un certain degré de programmation relativement à un objet intermédiaire (nous expliciterons ce degré de programmation lors de la définition du sous-type II⁻ du type II).

Schématisation :

(Cas de RECT et TACHE)



Ou encore :



• **QUESTIONS**

Quelles sont les conséquences pour l'utilisateur et l'activité mathématique des possibilités offertes par des logiciels de sous-type I-II (objet intermédiaire produit en mode commande - réponse produite en mode réponse) ?

Ces possibilités sont-elles divergentes ou convergentes par rapport à l'activité ?

LE TYPE II

Il s'agit de logiciels «commande».

La réponse au problème traité s'effectue en mode commande.

On peut distinguer :

Le sous-type II⁻

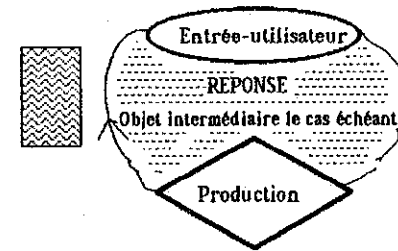
Toutes les entrées s'effectuent en mode commande.

Il n'y a pas de mode différé.

La commande de type pilotage permet néanmoins l'expression d'un algorithme pour résoudre un problème ; même si l'anticipation n'est pas une donnée immédiatement attachable au logiciel, il est permis de penser qu'elle peut intervenir. On peut parler d'un certain degré de programmation relativement à la réponse au problème traité (et, le cas échéant à un objet intermédiaire).

Schématisation :

(Cas de PICTOR)



Le sous-type II⁺

Toutes les entrées s'effectuent en mode commande.

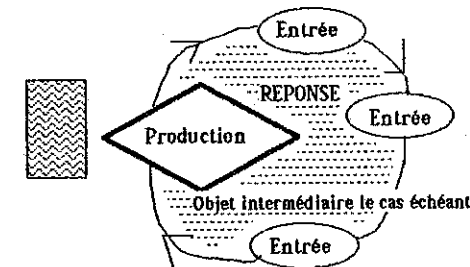
Il y a une possibilité de travail en mode différé.

Le degré de programmation possible est maximum.

L'anticipation est une donnée directement attachable au logiciel.

La schématisation est celle du sous-type II⁻, ou encore pour rendre compte de l'existence du mode différé :

(Cas de LOGO)



Avec les sous-types II⁻ et II⁺, il n'y a pas d'instructions de parcours.

Le sous-type II.I

Il s'agit de l'ancien sous-type III-II, que nous rebaptisons II-I pour rendre compte du fait que la réponse au problème traité s'effectue en mode commande, ce qui est caractéristique du type II.

Il y a donc sur le plan de la donnée de réponse les possibilités évoquées en sous-type II⁻.

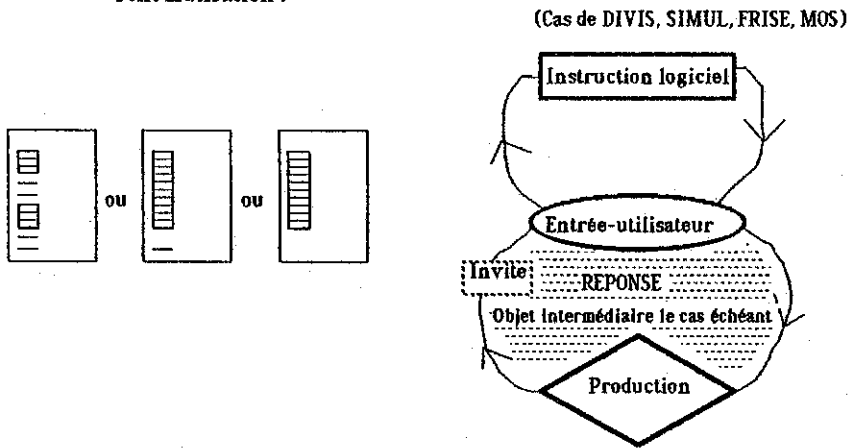
Certains éléments du type I interviennent, qui ont été mis à jour précédemment (formulation de consignes, ...).

Tout ceci explique la dénomination II-I (sous-type II à tendance I).

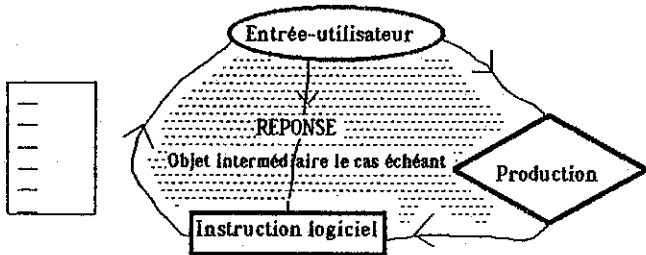
Les entrées de l'utilisateur se font essentiellement en mode commande.

Pour les logiciels étudiés, les entrées relatives à des branchements ont généralement statut de commande (instructions de parcours type c) ; mais, elles peuvent aussi avoir statut de réponse (c'est par exemple le cas pour MOSAÏQUE).

Schématisation :



Ou encore :



Les logiciels de sous-type II-I présentent des instructions de parcours.

• SYNTHESE ET QUESTIONS

Les types I, II, III ont été remodelés en une typologie à deux grands types avec plusieurs sous-types correspondant aux précédentes classes de la typologie.

Le tableau de correspondance s'établit ainsi :

Dénomination avant étude du statut des entrées-utilisateur	I	III		II		
		III-I	III-II			
Dénomination après étude du statut des entrées-utilisateur	I-I	I-II-	I-II+	II-I	II-	II+
		I		II		
Classement selon la communication des logiciels analysés	Puce	Rect Tache	Divis Simul Mos Frise	Pictor	Logo	

Ce tableau traduit un choix de notre part, qui consiste à privilégier le critère "statut des entrées utilisateurs". Les classes sont inchangées, mais par contre, leur regroupement l'est.

Le critère "statut des entrées utilisateur" nous semble fondamental.

En particulier, pourra-t-on parler d'attitudes, de démarches différentes de l'utilisateur, suivant le type dont relève le logiciel qu'il emploie ? Quelles seront les conséquences pour l'activité mathématique ?

Certains logiciels permettent la construction de la réponse en mode commande.

Mais s'agit-il toujours de la construction de la réponse ? De la construction d'un objet mathématique ? Quels rapports y-a-t-il entre ce que l'utilisateur construit, et le(s) concept(s) mathématique(s) en jeu ?

Si l'utilisateur peut « programmer », quand et comment apprend-il ? Y-a-t-il toujours réellement programmation ? Est-ce au bénéfice de l'activité mathématique ?

4- DEFINITION DE LA TYPOLOGIE COM

4a Descriptif de la typologie COM

L'étude des résultats fournis par l'analyse a priori des neuf logiciels selon la grille COM permet d'établir une typologie.

Celle-ci comporte deux types : les types I et II.

Chaque type comporte des sous-types.

Les sous-types I-I du type I et II⁺ du type II sont des pôles de la typologie COM.

La description de chaque sous-type comporte une schématisation relative à la structure des logiciels. Le principe de cette schématisation est donné en annexe V et dans le fascicule d'accompagnement.

Le type I

Ce type comporte deux sous-types (I-I et I-II), le sous-type I-II se décomposant lui-même en deux sous-types (I-II⁻ et I-II⁺).

Caractéristiques générales du type I

Les logiciels de ce type sont des logiciels «réponse».

La réponse au problème traité s'effectue en mode réponse.

L'utilisateur ne dispose pas d'instructions qui lui permettent d'exprimer un algorithme relativement à la réponse, d'anticiper, de programmer.

Il y a évaluation par rapport à l'attendu conceptuel.

Caractéristiques spécifiques au sous-type I-I

Avant l'emploi, l'utilisateur ne dispose de rien.

Les instructions sont données par le logiciel à l'écran.

Elles peuvent être de plusieurs genres, à l'exclusion du genre instruction de production.

En règle générale, les entrées utilisateurs ne sont pas des instructions. Elles ne peuvent l'être qu'exceptionnellement et concernent alors exclusivement le parcours du logiciel.

L'utilisateur est toujours en mode réponse, qu'il s'agisse de produire la solution du problème ou un objet intermédiaire. Il n'y a souvent qu'une seule possibilité par entrée-utilisateur, exigeant une construction personnelle de l'utilisateur (n'employant pas de données brutes du

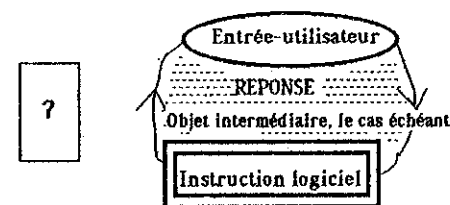
logiciel ou de son environnement).

Suite à une entrée, l'utilisateur reçoit une instruction.

Dans ce cadre, l'utilisateur ne décide rien.

Il n'y a aucune possibilité d'expression d'algorithme, d'anticipation, de programmation.

Schématisation



Caractéristiques spécifiques au sous-type I-II

Avant l'emploi, l'utilisateur peut disposer de renseignements.

Les instructions sont données à l'écran ou sur un document.

Elles peuvent être de plusieurs genres : le genre parcours est toujours présent ; le genre production également, mais il concerne la construction d'un objet intermédiaire.

Si celui-ci est toujours optionnel, on parlera du **sous-type I-II⁻**.

Si celui-ci est obligatoire sur au moins une partie du logiciel, on parlera du **sous-type I-II⁺**.

Il existe au moins un groupe d'instructions valides sur une partie du logiciel, ou des groupes d'instructions valides chacun à des endroits différents du logiciel.

Certaines entrées-utilisateur sont des instructions.

L'utilisateur est en mode commande de type pilotage pour la réalisation d'un objet intermédiaire et en mode réponse pour la réponse au problème traité.

Suite à une entrée, il y a production et/ou instruction.

Dans ce cadre, localement, l'utilisateur peut disposer d'éléments et

décider.

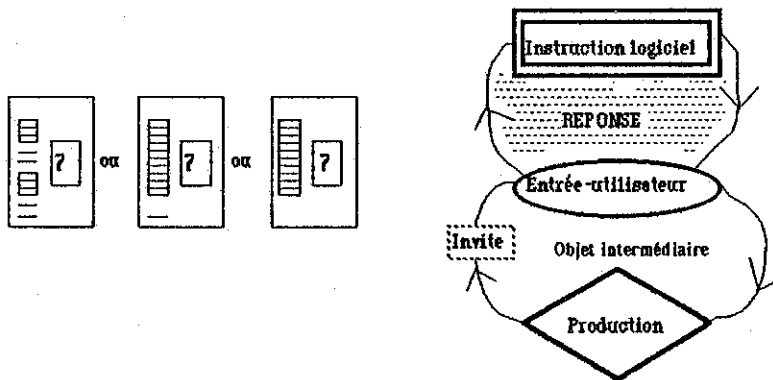
Il y a possibilité d'exprimer un algorithme, d'anticiper, de programmer pour la réalisation d'un objet intermédiaire. Néanmoins, l'anticipation n'est pas une donnée directement attachable au logiciel : il n'y a pas de mode différé.

Schématisations

On peut réaliser deux sortes de schémas pour le sous-type I-II.

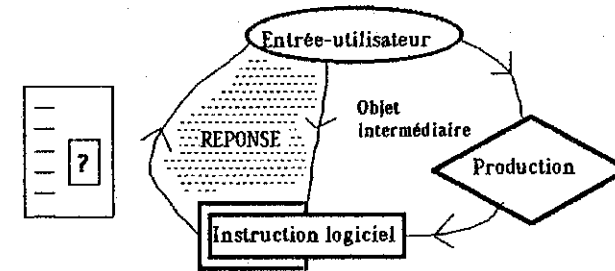
Notons qu'en sous-type I-II⁺, les deux genres de cycles sont nécessairement employés avec dominance des cycles production. En sous-type I-II⁻, les deux genres de cycle ne sont pas nécessairement employés ; il y a dominance des cycles réponse.

— Schéma a



Remarque : l'instruction invite, si elle est présente, est la même sur plusieurs cycles.

— Schéma b



Le type II

Ce type comporte trois sous-types (II-I, II⁻ et II⁺).

Caractéristiques générales du type II

Les logiciels de ce type sont des logiciels «commande».

La réponse au problème traité s'effectue en mode commande. Avant l'emploi, l'utilisateur peut disposer de renseignements. Il n'y a pas d'évaluation par rapport à l'attendu du conceptuel.

Dans le cadre précisé par chaque sous-type, globalement ou plus ou moins localement, l'utilisateur dispose d'éléments et décide.

Il y a possibilité d'exprimer un algorithme, d'anticiper, de programmer pour la réponse au problème traité ou la réalisation d'un objet intermédiaire.

Caractéristiques spécifiques aux sous-types II⁻ et II⁺

Les instructions sont données par un document.

Elles sont d'un seul genre : instructions de production.

Elles sont valides sur tout le logiciel.

Les entrées utilisateur sont des instructions.

L'utilisateur dispose toujours d'un mode commande de type pilotage.

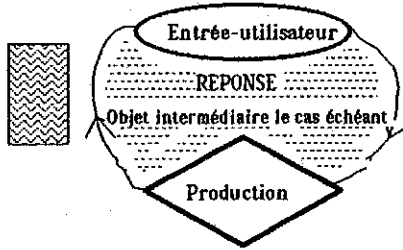
Suite à une entrée, l'utilisateur ne reçoit pas d'instructions.

• Caractéristiques spécifiques au sous-type II⁻

Suite à une entrée, il y a toujours production.

L'anticipation n'est pas une donnée directement attachable au logiciel : il n'y a pas de mode différé.

Schématisation



• **Caractéristiques spécifiques au sous-type II***

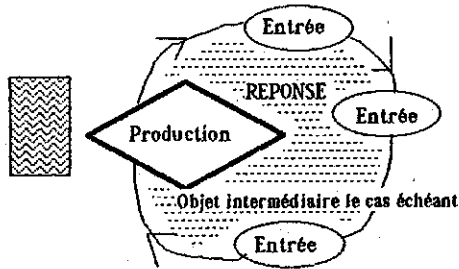
Il existe un mode différé.

L'anticipation est une donnée directement attachable au logiciel.

Schématisation

La schématisation du sous-type II* relative au mode pilotage est celle du sous-type II.

Schématisation relative au mode différé



• **Caractéristiques spécifiques au sous-type II-I**

Les instructions sont données à l'écran ou sur un document.

Elles peuvent être de plusieurs genres : les genres parcours et production sont toujours présents.

Il existe au moins un groupe d'instructions valides sur une partie du logiciel ou des groupes d'instructions valides chacun à des endroits différents du logiciel.

En règle générale, les entrées utilisateur sont des instructions.

Elles ne peuvent être qu'exceptionnellement des réponses et concernent

alors exclusivement le parcours du logiciel.

L'utilisateur est toujours en mode commande de type pilotage pour la réponse au problème traité ou la réalisation d'un objet intermédiaire.

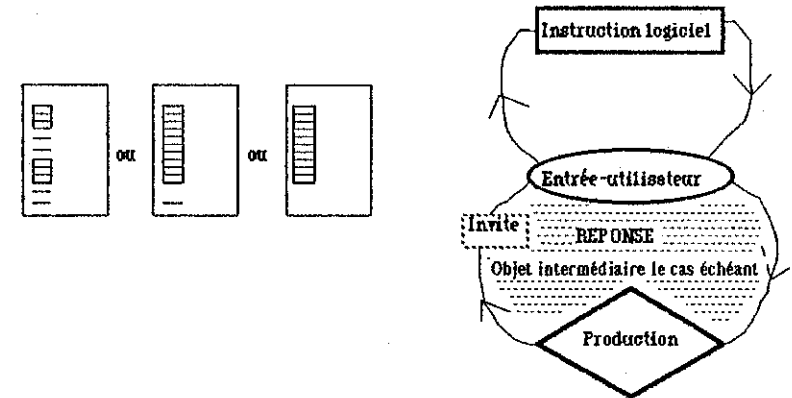
Suite à une entrée, il y a production et/ou instruction.

L'anticipation n'est pas une donnée directement attachable au logiciel : il n'y a pas de mode différé.

Schématisations

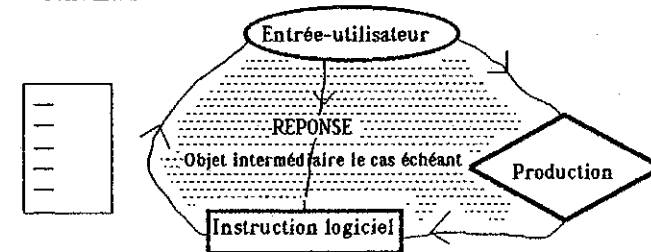
On peut réaliser deux sortes de schémas.

- Schéma a



Remarque : l'instruction invite, si elle est présente, est la même sur plusieurs cycles.

- Schéma b



4b Quelques remarques

– Les logiciels de type I sont opposés aux logiciels de type II pour le mode d'émission de la réponse au problème traité.

– Les logiciels de type I-II permettent ou exigent la réalisation d'un objet intermédiaire en mode commande à l'inverse des logiciels de type I-I.

– Les logiciels de type I-II localement, et surtout II-I, semblent présenter une structure "éclatement" de la structure II.

– La différence essentielle entre les sous-types (II⁺, II⁻) et II-I est l'existence en II-I d'instructions de parcours (avec les conséquences qui en découlent, dont la présence de certaines caractéristiques I).

On peut résumer ce qui précède par un arbre (voir page suivante) qui peut servir à effectuer un premier classement de logiciels.

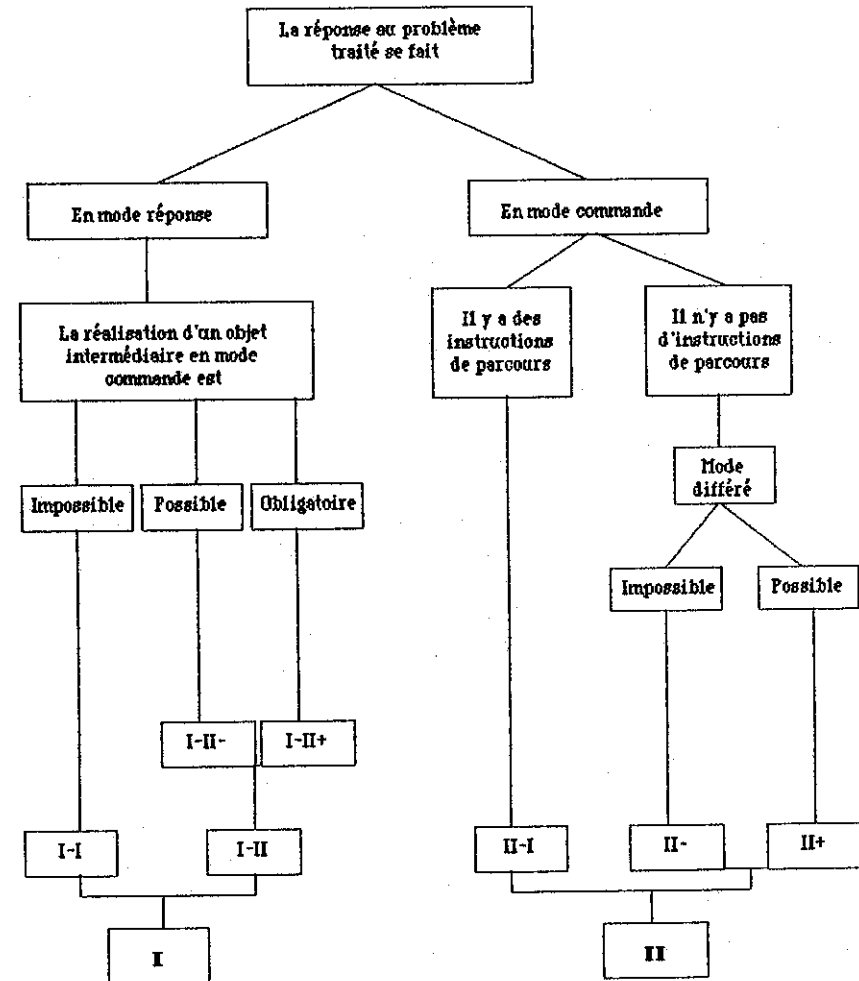
Pour assurer le classement, il conviendra de mener une étude précise.

– Un logiciel peut être globalement d'un certain type et emprunter des caractéristiques à un autre type : ainsi, pour le critère "évaluation par rapport à l'attendu conceptuel", certains logiciels de sous-type II-I fonctionnent en type I (redondance ?) tandis que certains logiciels de sous-type I-II fonctionnent en type II (manque ?).

D'autre part, un logiciel peut admettre un «sous-logiciel» d'un autre type que celui auquel il appartient.

– Dans un type donné, on peut établir une hiérarchie par analyse fine suivant les critères qui définissent le type.

Arbre de premier classement



4c Quelques questions

Nous rappelons que nous apporterons, au cours de la suite de l'exposé de la recherche, des éléments de réponse aux questions que nous allons maintenant poser.

- Comment l'utilisateur gère-t-il les divers genres d'entrée permis ?
- Quel est l'intérêt, eu égard à une activité mathématique, des instructions de production dont dispose l'utilisateur dans un logiciel ?
- Les logiciels de type II demandent à l'utilisateur de réaliser un apprentissage logistique.
 - Quand et comment l'utilisateur apprend-il à commander la machine ?
 - Quels rapports y-a-t-il avec l'activité mathématique en jeu ?
 - Quel est le statut de ce que l'utilisateur construit en mode commande ? S'agit-il d'un objet mathématique ?
- Les différences entre les types sont énoncés du point de vue de la communication proposée à l'utilisateur avec le micro-ordinateur. Recouvrent-elles d'autres différences ?
- Quels sont les enjeux de situations utilisant des logiciels des divers types ?

TYPOLOGIE GENERALE CONT/COM

Ce chapitre conclut la partie I de la recherche.

La typologie CONT propose un classement des logiciels du point de vue du contenu, de son organisation.

La typologie COM propose un classement des logiciels du point de vue de la communication avec le micro-ordinateur, de leurs formes.

A la fin de chacune des études établissant une typologie s'est posée la question de savoir si les différences entre les types recouvraient d'autres différences.

La comparaison des typologies CONT et COM a pour objet, si cela s'avère possible, d'apporter des éléments de réponse à cette question, de produire une typologie CONT/COM dite générale.

Les principaux résultats se retrouvent dans le fascicule d'accompagnement.

1- LIAISON DES TYPOLOGIES CONT ET COM

Les logiciels de référence se répartissent ainsi :

	PUCE	RECT TACHE	DIVIS SIMUL	MOS FRISE	PICTOR	LOGO
CONT	①		3.1	③ 3.2		②
COM	I-I	I	I-II+	II-I	II	II+

Nous rappelons que l'annexe IV présente le classement d'autres logiciels.

Par conséquent, nous dirons que :

- Les types 1 de CONT et I de COM se correspondent.

La typologie COM précise ici le type 1 de CONT par trois sous-types (I-I, I-II⁻, I-II⁺).

- Les types 2 et 3 de CONT et II de COM se correspondent.

De façon plus détaillée :

- Les types 3 de CONT et II-I de COM se correspondent.

La typologie COM précise ici le type II-I de COM par deux sous-types (3.1, 3.2).

- Les types 2 de CONT et II⁻, II⁺ de COM se correspondent.

La typologie COM précise ici le type 2 de CONT par deux sous-types (II⁻, II⁺).

2- CONCLUSION : DEFINITION DE LA TYPOLOGIE GENERALE

2a Préalable

Nous signalerons ici que les résultats ne semblent pas être strictement contingents :

- D'autres logiciels sont classés en annexe IV.

- En outre, notre point de vue de départ (considérer par une même analyse tous les logiciels, y compris les langages) et ses conséquences paraissent être globalement validés par l'évolution des logiciels ; nous expliciterons cette remarque au cours de la partie II de la recherche.

2b Descriptif de la typologie générale CONT/COM

L'analyse de la distribution des logiciels de référence dans les typologies CONT et COM conduit à conclure que :

- Les typologies CONT et COM ne sont pas contradictoires.

Les tendances dans une typologie vers un type "un" ou "deux" sont réaffirmées.

- La différence entre types dans une typologie recouvre d'autres différences.

Ceci confirme l'intérêt qu'il y a à considérer les deux points de vue (CONT et COM) : ils se complètent, voire se précisent.

- Il est possible de proposer une typologie générale en trois types avec sous-types.

Pour retrouver les caractéristiques de chaque type, il convient de se reporter à l'exposé des typologies CONT et COM qui figurent respectivement en pages 52 et 98, et dans le fascicule d'accompagnement : les chiffres en écriture arabe renvoient aux types de CONT, ceux en écriture romaine aux types de COM.

Le type A

- . sous-type 1 I-I
- . sous-type 1 I-II⁻
- . sous-type 1 I-II⁺

Le type B

- . sous-type 3.1 II-I
- . sous-type 3.2 II-I

Le type C

- . sous-type 2 II⁻
- . sous-type 2 II⁺

2c Classement des logiciels de référence selon la typologie générale

Les logiciels analysés se répartissent ainsi pour leurs caractères dominants :

Type A			Type B		Type C	
1 I-I	1 I-II-	1 I-II+	3.1 II-I	3.2 II-I	2 II-	2 II+
Puce		Rect Tache	Divis Simul	Frise Mosaïque	Pictor	Logo

3- QUESTIONS

La référence à un type donné du point de vue du contenu implique-t-elle la référence à un type donné du point de vue de la communication ? (et réciproquement ?).

La lecture des deux tableaux ci-dessous résumant la typologie générale autorise une réponse.

	Type A	Type B	Type C
CONT	1	3	2
COM	I	II II-I II- II*	

	Type A	Type B	Type C
CONT	1	3.1 3.2	2
COM	I-I I-II- I-II*	II-I	II- II*

Ainsi :

– On peut avoir le même type pour le contenu et des sous-types différents pour la communication (en A et C).

– On peut avoir le même type pour la communication et de sous-types différents pour le contenu (en B).

A contrario, au niveau des types mêmes, il semble qu'il y ait corrélation :

- Le type 1 de CONT est attaché au type I de COM.
- Les types 2 et 3 de CONT sont attachés au type II de COM ; plus précisément, le type 2 de CONT est attaché aux sous-types II de COM sans caractéristiques I, tandis que le type 3 de CONT l'est au sous-type de COM avec caractéristiques I.

La question générale reste : quels sont les enjeux de situations utilisant des logiciels des divers types ?

PARTIE II

LOGICIELS POUVANT IMPLIQUER

DES ACTIVITES MATHEMATiques

A L'ECOLE ELEMENTAIRE :

ENJEUX DIDACTIQUES



INTRODUCTION

La première partie de la recherche, répondant aux deux premiers objectifs, a établi une typologie qui confirme les hypothèses :

- La communication avec le micro-ordinateur est un élément spécifique.

- Il y a dépendance entre le type de contenu et le type de communication proposés par le logiciel.

La partie II s'intéresse aux trois derniers objectifs que nous rappelons ici :

- Confronter les différents types de logiciels dégagés dans la partie I et le vocabulaire informatique et didactique en usage.

- Emettre des hypothèses relatives aux enjeux de l'utilisation effective (pendant l'enseignement) de logiciels de chaque type, aux rôles impartis à la machine.

- Amorcer une prospective de construction de logiciels.

Une attention particulière sera portée à la communication en tant que variable didactique.

Nous rappelons que nos outils seront :

- Les résultats de la partie I de la recherche (critères d'analyse et description des types).

- Des observations que nous avons pu faire en situation d'enseignement.

- Les outils de notre cadre théorique, la didactique des mathématiques.

L'étude d'un glissement lors de la mise en œuvre d'un logiciel (logiciel d'un certain type utilisé dans un contexte attachable à un autre type) ne fait pas partie de nos objectifs.

Le plan adopté pour l'exposé vise à mettre en valeur les différences entre les types de logiciels obtenus.

Seuls les traits marquants sont envisagés.

Nous proposons une sorte d'étude panoramique qui pourrait constituer une base pour le développement d'une problématique de l'informatique-outil ; des travaux pointus, qui ne font pas partie du cadre de notre recherche, devraient compléter notre contribution.

Deux premiers chapitres apportent des compléments relatifs à la diversité des logiciels et à la communication : ils prolongent la partie I de la recherche, et introduisent l'examen des enjeux et des spécificités des logiciels selon une grille didactique, objet à proprement parler de la partie II.

Nous traitons ensuite successivement :

- des tâches de communication (nature et problèmes d'apprentissage).
- des tâches relatives au contenu de l'activité (définition de l'activité, statut de l'objet produit, validation).
- de la transposition de concepts mathématiques avec des logiciels.
- et enfin, en forme de conclusion, des spécificités des logiciels pour l'enseignement des mathématiques.

A propos du vocabulaire employé au cours de la partie II

Un logiciel de type X sera appelé "logiciel X".

Une communication de type X sera appelée "communication X".

Nous parlerons de logiciels X "stricts" pour désigner ceux de type X qui vérifient la totalité (ou presque) des critères du type; dans le cas contraire, nous emploierons l'appellation logiciel X "non strict".

UNE ORGANISATION POSSIBLE DE LA DIVERSITE DES LOGICIELS

1- INTRODUCTION

Le titre de ce chapitre rappelle un des objectifs d'ensemble de notre recherche.

Nous nous proposons ici de situer cette recherche pour l'étude de la diversité des logiciels.

Trois points de vue vont être mis en évidence :

- Apports possibles de l'analyse a priori (suivant les critères que nous avons définis).
- Apports possibles de la typologie générale.
- Apports possibles de l'ensemble des travaux à la mise au point d'une méthodologie d'évaluation des logiciels, et/ou de définition de séquences d'enseignement faisant intervenir un logiciel.

Notre approche initiale de la diversité des logiciels est un élément important pour l'ensemble du travail. Elle s'est faite conformément à notre position sur l'utilisation de moyens informatiques dans l'enseignement. C'est pourquoi, nous commencerons par l'exposé de cette position.

2- NOTRE POSITION SUR L'USAGE DE MOYENS INFORMATIQUES DANS L'ENSEIGNEMENT

Nous reprenons l'idée de Piaget selon laquelle "apprendre résulte d'agir".

Toutefois, nous nous intéressons à l'enseignement des mathématiques dans un contexte scolaire : l'apprentissage est à envisager dans son rapport avec l'enseignement. En particulier, la gestion des situations nous paraît un élément fondamental.

Nous ne développerons pas plus : le lecteur trouvera une expression détaillée de la position que nous venons d'esquisser dans les "Cahiers de didactique des mathématiques" (Publications IREM Paris VII), en particulier aux numéros 5 ("Quelques concepts, quelques généralités et quelques références"; collectif)

et 6 ("De la didactique des mathématiques à l'heure actuelle"; R. Douady).

Notre position relative à l'emploi de moyens informatiques dans l'enseignement se veut en accord avec la conception globale de l'enseignement que nous venons d'énoncer. Examinons les conséquences d'un tel préalable.

Nous ne pouvons adopter pour la sélection de logiciels une attitude du type de celle que nous avons décrite lors de l'exposé de l'approche usuelle des logiciels (voir p. 19-23). Nous ne nous satisfaisons pas d'une alternative non justifiée (LOGO VS didacticiels).

L'EPO ne correspond pas à notre choix pour l'apprentissage. Toutefois, nous ne trouvons pas a priori absurde que certains didacticiels (au sens courant) puissent intervenir utilement au cours de l'enseignement de concepts.

L'usage du langage LOGO est cohérent avec notre point de vue. Nous essaierons de montrer que ce logiciel a des spécificités autorisant une bonne gestion didactique. Nous ne l'acceptons pas pour autant comme solution universelle et n'adhérons pas au discours idéologique qui l'accompagne. Les gains en matière d'apprentissage mathématique ne semblent pas plus assurés avec LOGO, que les pertes par usage d'autres logiciels.

En résumé, nous refusons que les moyens informatiques fassent l'objet d'une discrimination non fondée. Nous situons l'ordinateur comme instrument didactique, l'activité avec celui-ci ne recevant pour signification que celle qui aura été définie par le maître avec la classe. Ainsi, l'emploi d'un logiciel doit être envisagé au même titre que celui d'un autre support didactique, c'est-à-dire pour sa pertinence dans la séquence projetée. Notamment, il doit être subordonné à une analyse des tâches, des enjeux. Tous les logiciels (LOGO, didacticiels, ...) sont concernés.

Par conséquent, nous admettons et utilisons le terme EAO au sens du rapport Schwartz, à savoir dès qu'intervient un usage, quel qu'il soit, de l'ordinateur pour l'enseignement. Le problème réside en l'évaluation des modalités de l'EAO. La recherche relatée ici, résultat d'une telle approche de la diversité des logiciels, constitue une tentative de définition de méthodes adaptées à la nouveauté du matériau.

3- APPORT DES GRILLES D'ANALYSE A PRIORI CONT ET COM

Un ensemble diversifié de logiciels est susceptible d'intervenir dans le cadre scolaire.

Une hypothèse implicite de la recherche est que leur analyse comparative, selon des critères identiques, doit être possible et peut s'avérer

fructueuse pour une meilleure connaissance du rôle des logiciels dans l'enseignement.

C'est pourquoi, nous envisagerons notamment par ces deux hypothèses que la suite de la recherche examine, l'apport des grilles d'analyse CONT et COM :

– *Les grilles CONT et COM permettent l'analyse a priori de tous les logiciels.*

Plusieurs utilisateurs l'ont attesté. Les produits de type tableur, traitement de texte, langage... ne sont plus exclus de l'examen comme ils l'étaient avec d'autres moyens ; l'annexe IV les classe.

Ceci nous paraît positif, compte-tenu de l'importance que nous attachons à une non-discrimination de départ. Une raison de cette aptitude particulière à l'analyse de tous les logiciels est exposée au paragraphe 4 de ce chapitre.

– *Les grilles construites visent à repérer la présence ou l'absence d'éléments spécifiques de gestion des situations d'enseignement (grille CONT) et de communication (grille COM).*

Elles interrogent selon des critères objectifs, bases pour une définition des tâches et des enjeux. Elles peuvent donc constituer des aides à la décision didactique.

4- APPORT DE LA TYPOLOGIE GENERALE

Aujourd'hui, nos hypothèses explicites de recherche se confirment : la double approche préconisée appliquée à plusieurs logiciels a permis d'établir deux typologies a priori, non contradictoires, se précisant l'une l'autre au sein d'une typologie générale, identificatrice de spécificités du double point de vue contenu, communication.

4a Développement d'une problématique de l'informatique-outil

La typologie élaborée peut constituer un instrument de développement d'une problématique de l'informatique-outil.

• D'une part, elle nous permet de contribuer à une approche didactique du problème de l'informatique-outil :

– Elle a donné naissance à une amorce de théorisation (cf partie I, rapports contenu/communication). Les résultats correspondants mettent en cause le contrat didactique usuel, en ce qui concerne l'organisation des rapports élève/système informatique ; nous y reviendrons.

– La mise en évidence de la nécessité de la communication élève-ordinateur consacre la réalité de nouvelles tâches pour l'utilisateur. Celles-ci ne devraient plus rester l'objet d'un contrat implicite : des propositions pour expliciter ce contrat sont faites au chapitre "Les tâches de communication".

– Notre typologie va aussi nous permettre d'émettre des hypothèses prédictives sur l'ensemble des tâches que les enfants rencontreront effectivement en faisant fonctionner les différents types de logiciels, et plus généralement, sur ce que l'on peut attendre (aussi bien positivement que négativement) des logiciels.

Cette formulation didactique, à fondement rationnel, des divers types de logiciels correspondants au regroupement de caractéristiques, constitue l'objet de la deuxième partie de la recherche et son apport à l'étude de la diversité des logiciels.

• D'autre part, il est possible d'apporter un élément de réponse à la question de la construction de logiciels pour l'enseignement : on peut utiliser la typologie "presque à l'envers", se donner des objectifs en termes de tâches ou d'enjeux, et en déduire des caractéristiques des logiciels à produire ; on satisfera ainsi une association inédite ou non de critères, dont on a précisé les enjeux.

Il convient toutefois d'examiner le degré de stabilité du travail que nous avons entrepris pour mieux cerner la validité de ces apports.

Deux groupes de questions se posent :

– Les critères qui interviennent dans les grilles d'analyse sont-ils pertinents face à l'évolution des logiciels? De nouveaux produits ne risquent-ils pas de résister à l'étude?

– Les types de la typologie ne seront-ils pas caduques rapidement? Quels remaniements faudra-t-il opérer? Qu'en sera-t-il des enjeux dégagés?

Nous répondrons en faisant plusieurs hypothèses :

– *A moyen terme, les critères semblent fiables et suffisants pour rendre compte des produits informatiques.*

– *A moyen terme aussi, les logiciels semblent pouvoir se classer selon les types répertoriés. Néanmoins, des produits originaux introduisent déjà des nuances, comme devrait le faire la prospective de construction que nous avons proposée.*

Les apports seraient donc valides au moins à moyen terme. D'autre part,

le vieillissement inévitable atteindra d'abord et essentiellement la typologie. C'est pourquoi, dans la suite du travail, nous exposerons les enjeux non par types de logiciel, mais par catégories d'enjeux en signalant à chaque fois les types de logiciels actuellement concernés. Des ajustements ultérieurs resteront donc permis.

À l'appui des hypothèses qui précèdent, signalons que, à notre connaissance, il ne semble pas que nos principes et résultats en matière d'étude des logiciels aient été mis en défaut, de 1982 à ce jour.

Ainsi, en Mars 1986, il est possible de classer les logiciels se réclamant de l'Intelligence Artificielle (globalement en type C), comme ceux apparus dans le cadre de l'opération "Informatique Pour Tous" (types divers).

En fait, depuis 1982, nous avons assisté, plus à une modification de la distribution des logiciels suivant les types de référence, qu'à une invalidation de la typologie. Au départ, la situation, reflétée dans une certaine mesure par la typologie, était bipolaire et déséquilibrée. Ainsi, peu de produits informatiques pouvant impliquer des activités mathématiques et ne prétendant pas au titre de "langage" se regroupaient autour du pôle à mode commande. À l'inverse, le pôle à mode réponse comptait de nombreux représentants. Aujourd'hui, la tendance est à un rééquilibrage des deux pôles et à une diversification des produits intermédiaires.

Il est probable que le parti pris d'origine «considérer tous les logiciels qu'ils soient d'exploitation ou d'application», qui ne s'attache pas à une différenciation en terme de distance à la machine (nombre de traducteurs), mais veut permettre une analyse fonctionnelle didactique est une raison de la stabilité relative.

Les critères sont fortement liés à des problèmes didactiques et, par conséquent, moins sujets au dépassement que d'autres dépendant du stade de développement d'une technologie.

En outre, l'étude, se situant du point de vue des utilisateurs, pouvait vraisemblablement appréhender une demande à laquelle certains produits commençaient à répondre implicitement.

En 1982, considérer des logiciels tels que PICTOR ou un tableur comme analogues sous bien des aspects aux langages n'était pas accepté. Actuellement, cette position est mieux admise.

Nous avons déjà cité en page 12 la nomenclature en vigueur dans le "Catalogue des logiciels Informatique Pour Tous" de Novembre 1985 qui rassemble langages, tableurs, ... sous le terme «outil logiciels». Ce changement de point de vue d'autrui qui entérine une identité que nous avons perçue, n'est pas sans rapport avec le développement et le succès hors secteur éducatif de logiciels à interface utilisateur garantissant un mode commande. Remarquons d'ailleurs qu'un tel mode commande, est encore

aujourd'hui mieux décelé par de nombreux interlocuteurs pour les logiciels traitement de textes, tableurs, ... que pour ceux du secteur strictement éducatif. Néanmoins, il est maintenant plus aisé de faire envisager sur la base des logiciels semi-professionnels évoqués le point de vue "mode commande".

Le contexte extra-scolaire, référence inconsciente pour la production de logiciels, peut également expliquer le rééquilibrage des pôles et la diversification des produits intermédiaires dont il a été question.

Nous poursuivrons en explicitant le fait qu'une approche faisant une trop large part à un état donné d'une technologie ne peut qu'être rapidement désuète.

La transformation et la multiplication des matériels sont des causes d'évolution des logiciels¹. En micro-informatique, les frontières s'estompent rapidement. Ainsi, matériel et logiciel ne sont plus des distinctions fiables. Ce qui était autrefois réalisé d'une seule façon peut l'être maintenant de diverses manières.

Par exemple, les produits d'exploitation de l'ordinateur (auxquels se rattachent les traducteurs de langage) ne sont plus que rarement construits en matériel; à l'inverse, des applications peuvent l'être (cartouches de mémoire morte diverses). Une même couche logicielle peut accueillir un interpréteur ou un tableur. Les applications ne sont pas nécessairement écrites sous un traducteur évolué. Tout ceci remet en cause la dichotomie ancienne "langage VS application" qui reposait pour beaucoup sur une réalisation à un niveau matériel des langages et sur un recours des applications à au moins un traducteur évolué.

Parallèlement, la déstabilisation du terme langage a été engagée par l'apparition de générateurs de programmes, de langages de commande, le cas échéant à interface graphique (icônes, pictogrammes) en liaison avec les performances accrues des matériels.

De plus, auparavant, on ne disposait que de logiciels spécialisés répondant à une fonction alors qu'aujourd'hui des logiciels dits intégrés assurent la mise en relation de plusieurs genres d'applications.

Le vocabulaire, dont on connaît l'importance pour l'abord du problème de l'intervention des logiciels dans l'enseignement, se périmé donc vite, en même temps qu'évolue le contexte micro-informatique. Les réseaux locaux ou télématiques transformeront vraisemblablement aussi le panorama actuel.

En conséquence, distinguer des logiciels par un vocabulaire se référant à un domaine d'application (didacticiel), au nombre de fonctions assurées (progiciel ou logiciel intégré), à une technologie est insuffisant, voire déjà caduque. En outre, cela est limitatif: si l'état donné du développement technologique régit seul la demande logicielle éducative, des attentes ne seront pas formulées, car correspondant à des produits difficilement ou complètement inconcevables. Elles pourraient l'être si la référence est d'ordre didactique (sous une hypothèse de réalisation à plus ou moins long terme).

1. Sur l'évolution des logiciels, le lecteur pourra consulter:
 - "L'informatique, mode d'emploi" de B. Lorimy (Ed Fayard).
 - La revue "Pour la science", numéro spécial n°85 (Novembre 1984).

La naissance du langage PROLOG est significative à cet égard : elle correspond à une libération de la pression technologique d'un temps.

Citons A. Colmerauer dans la préface de l'ouvrage "Prolog" :
 "Prolog est né d'un pari: créer un langage de très haut niveau, même inefficace au sens des informaticiens de l'époque. L'efficacité consistait alors à faire exécuter très rapidement par une machine, des programmes écrits laborieusement. Le pari était donc de pouvoir écrire rapidement des programmes, quitte à ce que la machine les exécute laborieusement. Une fois libérés de cette obsession d'efficacité, on a pu faire appel aux formalismes et aux mécanismes d'inférences de la logique mathématique, mécanismes inefficaces mais tellement puissants!"

Bien plus modestement, nous nous souvenons avoir souhaité l'intégration au logiciel PUCE de l'option CALCULATRICE dans un esprit qui n'est pas sans rappeler les "accessoires de bureau" disponibles maintenant de façon courante sur certains ordinateurs.

4b Vocabulaire

La typologie élaborée peut aussi permettre de redonner un sens au vocabulaire existant, à moins qu'elle ne serve à la mise en place de nouveaux termes.

Nous sommes conscient des problèmes liés au vocabulaire ; cependant, il nous a semblé intéressant de tenter sa clarification.

Une association des termes existants aux types que nous avons dégagés peut être réalisée, en s'appuyant d'une part sur le classement de logiciels selon nos critères, et d'autre part sur la dénomination que autrui leur attribue. L'appariement que nous soumettons paraît valide actuellement. Il n'en reste pas moins conjoncturel. En effet, il conviendrait de le reconsidérer si la terminologie (didacticiel, imagiciel, EAO, ...) est conservée pour des logiciels originaux, c'est-à-dire pour des logiciels qui, ne remettant pas en cause le propos de la recherche dont nous avons postulé la stabilité au moins à moyen terme, satisferaient un autre regroupement de critères que les produits, bases de l'appariement.

A la lecture du tableau de synthèse concernant la terminologie que nous proposons en page 121 et dans le fascicule d'accompagnement, on remarque qu'un même terme du vocabulaire d'usage peut s'appliquer à des logiciels de type différent : ceci confirme que la diversité des produits informatiques peut être masquée par la terminologie.

TERMINOLOGIE

D'après le vocabulaire d'usage		D'après la typologie générale	
EPO	Didacticiels OCM	Type A 1 I-I	Logiciels réponse Pas de langage Mini langage de commande disponible pour objet Intermédiaire
	Didacticiels	1 I-II ⁻	
	Didacticiels	1 I-II ⁺	
EA	Didacticiels Imagiciels	Type B 3.1 II-I	Logiciels commande Langage de commande
	Imagiciels Logiciels-outils	3.2 II-I	
O	Logiciels-outils Progiciels Logiciels intégrés	Type C 2 II-	Langage de programmation
	Tableaux Traitements de textes Gestionnaires de données		
Projet LOGO Micromondes	Langage auteur Langages de programmation Générateurs de programmes	2 II ⁺	

EAO

Micromonde
démarche

Micromonde
objet

Le vocabulaire existant reprend un sens fonctionnel en référence à la définition des types¹ et aux enjeux que nous mentionnerons par la suite. Toutefois, il est possible de lui en préférer un autre, plus descriptif des spécificités, qui intégrerait les deux angles d'analyse (logiciel à mode commande, à mode réponse, ouvert ou non pour l'objet du problème et/ou la démarche, ...).

Nous terminerons le commentaire du tableau récapitulatif en précisant les termes « langage de commande » et « langage de programmation ».

Pour cela, nous évoquerons à nouveau stabilité à moyen terme et évolution des logiciels : les différences entre certains des sous-types pourraient ne pas exister, ou plus exactement, il serait possible de concevoir des logiciels ne les faisant pas apparaître.

Ainsi, la différenciation entre les sous-types II-I et II⁻ de la communication de type II provient de l'existence en sous-type II-I d'instructions de parcours (type repérage). Ces instructions, qui peuvent paraître nécessaires (le logiciel décide du problème à traiter) ne sont toutefois pas indispensables. Les "langages" graphiques (par icônes ou pictogrammes) éliminent les instructions de parcours-repérage en affichant directement les commandes disponibles. En l'absence d'interface graphique, on pourrait aussi penser à transmettre des éléments de syntaxe élémentaire sur un document.

Lors de l'usage de MOSAÏQUE, les instructions de repérage étaient vite devancées par les enfants qui commandaient, sans attendre leur affichage.

Si, ultérieurement les sous-types II-I et II⁻ se confondent, qu'en sera-t-il des types B et C ? Peuvent-ils être assimilables à un seul type ? Comporteraient-ils un mode différé comme en sous-type II⁺ ? Un mode différé nous intéresse-t-il et en quoi ?

En sous-type II⁻, il peut exister des possibilités d'enregistrement et de rappel d'une production constituée en fichier de données, mais l'utilisateur ne peut produire un fichier programme, un logiciel, c'est-à-dire un objet exécutable, comme en II⁺.

1. Seule, la caractéristique du point de vue de la communication est indiquée succinctement dans le tableau ; la relecture du descriptif des types renseignera plus précisément.

C'est ainsi que nous distinguons :

— Les logiciels avec « langage de programmation » (y compris les générateurs de programmes) appartenant au sous-type II⁺, qui permettent la création de fichiers programmes.

— Les logiciels avec « langage de commande » appartenant au sous-type II⁻, qui permettent la création de fichiers de données, mais non celle de fichiers programmes.

Toutefois, cette distinction pourra-t-elle se justifier longtemps ... ?

5- CONCLUSION : EVALUATION DES LOGICIELS

Nous rappelons que nous séparons l'analyse d'un logiciel de son évaluation et retenons le principe d'une évaluation en trois étapes :

- analyse a priori du logiciel (pour le contenu et la communication), indépendamment de son utilisation effective pendant l'enseignement, sans émission d'hypothèses relatives aux caractéristiques et enjeux de situations qui l'emploieraient.

- utilisation des résultats de l'analyse a priori pour cerner tâches et enjeux didactiques.

- confrontation des résultats de l'analyse a priori aux acquis didactiques sur le sujet mathématique en question; rejet ou adoption du logiciel dans le cadre de séquences; expérimentation.

Par conséquent, selon ce protocole, l'évaluation de logiciels comme la définition de séquences d'enseignement les faisant intervenir, repose sur l'ensemble des travaux de notre recherche.

Nous soulignons aussi l'indispensable confrontation aux résultats didactiques connus.

Le genre d'utilisation d'un logiciel retenu pour l'élaboration d'une séquence, les caractéristiques de la séquence devraient dépendre du type de référence du logiciel, conformément à ce que nous exposerons par la suite. Nous redirons à ce propos que l'étude d'un glissement lors de la mise en œuvre (logiciel d'un certain type utilisé avec un contexte attachable à un autre type) ne fait pas partie des objectifs de cette recherche.

Nous terminerons par une interrogation.

Nous disposons de critères dont nous allons déterminer les enjeux didactiques. Nous avons déjà proposé que la réalisation future de logiciels satisfasse une association consciente, inédite ou non de critères.

Dès lors, une question fondamentale subsiste : quels aspects de quelles questions mathématiques peut-on transposer selon tel regroupement de critères ?

LOGICIELS ET COMMUNICATION COMPLÉMENTS

La première partie de la recherche a établi que les logiciels font référence à deux types de communication : le type I fondé sur le mode réponse (cycles «entrée utilisateur, instruction machine») et le type II basé sur le mode commande (cycles «entrée utilisateur, production»). Ceci paraît s'appliquer à l'ensemble des logiciels existants. En particulier, les logiciels attachés au domaine de l'intelligence artificielle (langage Prolog, systèmes experts ...) ne dérogent pas à cette règle : même si d'autres spécificités peuvent leur être reconnues, nous les situons en type II.

Nous formulerons cette hypothèse :

Les deux seuls types de communication envisageables à moyen terme pour les logiciels sont les types I et II.

Pour faciliter la lecture des chapitres qui suivent, nous exposerons quelques précisions relatives au type II de communication, accompagnées de commentaires introduisant des développements ultérieurs.

1- RAPPORT ENTRE LES INSTRUCTIONS DE PRODUCTION DES LOGICIELS A COMMUNICATION II ET LES PRODUCTIONS QU'ELLES ENGENDRENT

Nous voulons éclaircir ici le rapport entre ce que l'utilisateur exprime et ce qu'il obtient, lorsqu'il utilise une instruction de production.

1a Logiciels à sémantique déclarative

Il s'agit essentiellement de logiciels issus du domaine de l'intelligence artificielle (le langage PROLOG, un système expert, par exemple). Nous ne traiterons de leur cas que pour quelques remarques.

Les remarques faites dans le paragraphe -1a- à propos des logiciels à sémantique déclarative ne s'appliquent pas aux jeux de cible conçus spécialement pour l'école élémentaire. Les logiciels de jeux de cible sont analysables comme logiciels à

sémantique déclarative. On peut les assimiler à de mini systèmes-experts.

Bien que nous n'étudions pas ces logiciels dans notre recherche, en raison de leur apparition récente, nous dirons que les instructions qu'ils offrent, permettent de recenser des caractéristiques de l'objet recherché. Contrairement aux logiciels à sémantique procédurale, où comme nous le verrons l'utilisateur exprime une description, avec ces logiciels il en obtient une (le cas échéant, partielle) tout en la recherchant : ceci ouvre un champ qui nous paraît très prometteur.

Les instructions de production proposés par les logiciels à sémantique déclarative servent à poser le problème en terme de relations, ou encore à obtenir des éléments de sa description permettant de le résoudre. On dispose d'un formalisme – que n'offre pas un logiciel conventionnel –, que certains considèrent proche de la pensée usuelle.

Si ceci nous semble être une originalité féconde, nous nous demandons si l'emploi de tels logiciels peut se concevoir à l'école élémentaire. L'enfant de cet âge est-il en mesure d'émettre des descriptions relationnelles pour des objets qu'il ne maîtrise pas? Pour quelles questions? Les rapports que nous avons consultés portent sur des problèmes tous similaires (essentiellement, l'étude de l'ensemble des parallélogrammes). Une bonne connaissance des objets et des relations qu'ils entretiennent paraît nécessaire. En outre, l'enfant peut-il employer une syntaxe complexe (par exemple, Prolog s'appuie sur la logique des prédicats du premier ordre et exige des déclarations en conséquence)? Nous espérons que des logiciels et des recherches répondront à ces interrogations. A cette date, compte-tenu des remarques qui précèdent, nous ne développerons que le cas des logiciels ayant une sémantique procédurale.

1b Logiciels à sémantique procédurale

Le langage de commande ou de programmation propose ici à l'utilisateur des instructions de production pour une description intrinsèque, procédurale non relationnelle, de problèmes. Celle-ci peut être d'un statut différent suivant le logiciel, pouvant aller jusqu'à l'explicitation des tâches que devra accomplir l'ordinateur.

Nous préciserons d'abord le rôle des instructions de production. Elles autorisent l'utilisateur à montrer (afficher à l'écran, dessiner au sol, faire entendre, ...) un objet O ayant un rapport avec un objet arithmétique ou géométrique O'.

Plusieurs cas peuvent se présenter:

– les objets O et O' sont du point de vue de l'affichage identiques. Toutefois, l'objet O produit grâce au langage de commande ou de programmation par l'enfant peut ne pas avoir directement statut

d'objet mathématique (ou un statut incorrect): il peut n'être, par exemple, qu'un graphisme. C'est pourquoi, nous le distinguons de son homologue O' , avec statut mathématique.

- la question proposée à l'enfant peut l'amener à traiter un problème de type stratégique sur le domaine arithmétique, logique, ou géométrique. Dans ce cas, l'objet O affiché peut constituer un élément de la stratégie sans être directement O' , objet explicite de la question.

Ces remarques expliquent notre distinction entre O et O' . Elles soulignent aussi que les objets affichés grâce aux instructions peuvent participer à la résolution du problème, sans l'être. Ceci sera revu dans les chapitres qui suivent.

Plusieurs objets O_1, O_2, \dots, O_n peuvent concourir à la réalisation de O .

Ainsi par exemple, on pourra construire un rectangle (O_1) et un cercle (O_2), pour obtenir avec PICTOR cet objet (O):



L'examen des logiciels à communication II permet de caractériser plusieurs genres de systèmes d'instructions de production d'un objet O_i que nous désignerons par S_1, S_2, \dots, S_p . Nous avons ainsi établi une liste de huit systèmes d'instructions, qu'il conviendra vraisemblablement de compléter ultérieurement.

LES HUIT SYSTEMES D'INSTRUCTIONS DE PRODUCTION

Les instructions qui assurent la production d'un objet O_i impliqués par un objet O correspondent à :

□ l'objet O_i .

$S_1 = [O_i]$

Ainsi avec le logiciel LOGO, complété d'une procédure prédéfinie RECTANGLE, l'objet O_i "rectangle" peut s'obtenir par l'instruction RECTANGLE, c'est-à-dire directement l'instruction "objet O_i ".

L'instruction "objet O_i " signifie construire l'objet O_i désigné par l'instruction même (et en conséquence désigné par l'utilisateur), avec des paramètres définis par le logiciel. Elle comprend donc implicitement tout ce qui est nécessaire à la construction de l'objet, ceci étant totalement en charge du logiciel. La réalisation est transparente pour l'utilisateur; il n'a pas à décrire les tâches nécessaires à la production. Les paramètres de l'objet peuvent néanmoins lui apparaître, suite à son affichage.

□ un ou plusieurs paramètres p_i de l'objet O_i . $S_2 = [p_i]$

Ainsi avec le logiciel JOLIGONES, pour obtenir un polygone convexe régulier, on peut donner comme instructions deux de ses

paramètres, par exemple le nombre de ses côtés et la mesure commune de ses angles.

Les instructions "paramètre p_i " signifient construire avec ces paramètres désignés par l'utilisateur un objet O_i défini par le logiciel.

□ l'objet O_i et certains de ses paramètres. $S_3 = [O_i, p_i]$

Ainsi avec le logiciel PICTOR, pour obtenir un rectangle aux côtés parallèles aux bords de l'écran, on peut fournir comme instructions "objet rectangle" que l'on désigne sur la barre du menu, et deux de ses paramètres que sont deux sommets opposés que l'on pointe sur l'écran.

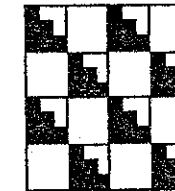
Avec le logiciel LOGO et une procédure à paramètres (longueur, largeur) pour la réalisation d'un rectangle, l'instruction RECTANGLE 3 12 revient pour l'utilisateur à désigner l'objet et deux de ses paramètres. On pourrait aussi citer dans ce cadre les logiciels permettant la représentation graphique de fonctions lorsqu'on peut en désigner une parmi plusieurs; dans le cas contraire, le logiciel propose un système de type S_2 .

L'explicitation du système S_3 est identique à celle du système S_2 , avec cette différence: l'objet O_i est ici désigné par l'utilisateur.

Remarque: Les systèmes S_1, S_2, S_3 recouvrent toutes les possibilités de combinaison à partir des données O_i, p_i .

□ un ou des sous-objets Ob_i de O_i en même temps que des opérations implicites Op_i sur ceux-ci. $S_4 = [Ob_i, Op_i]$

Ainsi avec le logiciel MOSAIQUE, pour produire l'objet O_i ci-dessous, on peut donner comme instructions "sous-objet motif" (avec des paramètres que sont couleur du fond et du motif), ce qui inclut des opérations de translation sur ce dernier.



Objet O_i



Sous-objet

L'instruction "sous-objet Ob_i " signifie, construire avec le sous-objet désigné et des opérations implicites un objet O_i défini par le logiciel.

Elle comprend donc implicitement la donnée d'opérations définies par le logiciel et attachées au sous-objet. Celles-ci peuvent apparaître à l'utilisateur

lors de l'affichage, leur réalisation lui restant transparente.

des opérations Opi en même temps que des sous-objets implicites Obi sur lesquels elles s'appliquent. **S5-[Opi,Obi]**

Ainsi avec le logiciel DIVIS, on peut donner l'instruction "opération -5" qui s'appliquera à un stade donné sur le sous-objet constitué du nombre résultant des soustractions successives effectuées jusqu'alors.

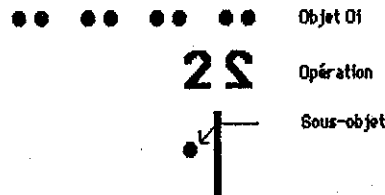
Avec le logiciel SIMUL, dans le contexte de commandes C (affichage par colonne) et M (affichage maîtrisé par le choix d'une option I), la sélection de la barre d'espace comme choix I est une instruction "opération" qui s'applique sur le sous-objet constitué par la grille partielle présente à l'écran.

L'instruction "opération Opi" signifie effectuer l'opération désignée ainsi par l'utilisateur sur les sous-objets en place, ceux-ci étant implicites.

un ou des sous-objets Obi de Oi et des opérations Opi relatives à Oi. **S6-[Obi, Opi]**

Ainsi, avec le logiciel SIMUL, l'affichage de la grille peut se faire grâce aux instructions "sous-objet ligne L" et "opération P" d'affichage continu.

Avec le logiciel FRISE, pour produire l'objet Oi ci-dessous, on peut donner les instructions "opération" et "sous-objet" indiquées.



L'explicitation du cas S6 est analogue à celle du système S4, avec cette différence: ici, les opérations Opi, comme les sous-objets, sont désignées par l'utilisateur.

des sous-objets Obi de l'objet Oi. **S7-[Obi]**

Ainsi avec le logiciel LOGO, pour produire l'objet Oi ci-dessous, on pourra donner les trois instructions sous-objets: "segment (orienté)" [AV 15], "secteur angulaire (orienté)" [TD 90], "segment (orienté)" [AV 10].



Avec le logiciel PICTOR, pour produire un objet Oi rectangulaire, on peut, ayant sélectionné le mode dessin, donner les instructions "sous-objet" que sont les désignations directes à l'écran des points du rectangle souhaité.

L'instruction "sous-objet" signifie strictement construire le sous-objet de Oi désigné ainsi par l'utilisateur. Elle n'inclut pas d'opérations implicites. L'utilisateur définit et produit l'objet Oi par emploi itéré d'un système S7 associé, le cas échéant, à un système de type S8. Avec ce type de système d'instructions, il doit expliciter à la machine les tâches inhérentes à la production de l'objet Oi.

des opérations Opi sur des sous-objets de Oi préalablement mis en place par l'utilisateur. **S8-[Opi]**

Ainsi avec le logiciel PICTOR, pour produire un rectangle plein, on peut faire afficher tout d'abord à l'écran un rectangle vide (sous-objet), puis commander l'"opération" de remplissage en désignant son pictogramme sur la barre du menu et un point intérieur à la figure.

Les opérations numériques du logiciel LOGO constituent un autre exemple.

On notera qu'ici la désignation de l'opération ne comprend pas implicitement celle des objets sur lesquels elle s'applique — ce qui était le cas avec un système de type S5.

En conséquence, ici, la production d'un objet Oi relève de l'utilisation d'un des systèmes précédents (en particulier S7) conjointement à un système S8. Toutefois, il ne s'agit pas du contexte S6: les opérations ne sont pas attachées à des objets dans un but défini par le logiciel, mais à des classes (les opérations numériques aux nombres, les remplissages aux domaines fermés, etc ...), sans objectif prédéfini par le logiciel.

2- TYPOLOGIE ET INSTRUCTIONS DE PRODUCTION DES LOGICIELS A COMMUNICATION II

Les exemples cités montrent que la production d'un objet Oi avec un logiciel donné peut faire intervenir différents types de systèmes Sk de production.

Par conséquent, nous n'attacherons pas nécessairement un unique système Sk à un logiciel à communication II.

Par la suite, nous préciserons les relations entre les systèmes Sk et les

sous-types de la communication II.

Ici, nous nous contenterons de proposer un tableau de synthèse regroupant les cas évoqués :

Système	Logiciels	Sous-type	Type
S1={O _i }	Macroprocédures LOGO sans paramètres	2 II+	C
S2={p _i }	JOLYONES	3.2 II-I	B
S3={O _i , p _i }	PICTOR; Macroprocédures LOGO avec paramètres	2 II ⁻ ; 2 II+	C
S4={Ob _i , Op _i }	PUZZLE; MOSAÏQUE	3.1 II-I; 3.2 II-I	B
S5={Op _i , Ob _i }	DIVIS; SIMUL	3.1 II-I	B
S6={Ob _i , Op _i }	FRISE; SIMUL	3.1 II-I; 3.2 II-I	B
S7={Ob _i }	LOGO; PICTOR	2 II ⁻ ; 2 II+	C
S8={Op _i }	LOGO; PICTOR	2 II ⁻ ; 2 II+	C

3- FONCTION DES INSTRUCTIONS DE PRODUCTION ; CONCLUSION

Nous dirons que les instructions de production (systèmes S_k) des logiciels à communication II et à sémantique procédurale correspondent à des objets ou opérations primitifs permettant la construction d'objets complexes (O_i) par une description¹.

Nous n'aborderons ici que quelques conséquences de la mise à jour des systèmes d'instructions qui seront développées parmi d'autres au cours des chapitres suivants. Ainsi, nous avons employé à dessein le qualificatif "primitif" et le terme "description" en étant conscient de leurs multiples sens dans ce contexte, ce dont nous allons nous expliquer.

La définition des différents systèmes S_k incite à penser que le degré de primitivité peut varier. On peut tenter de juger de celui-ci en référence à l'importance de la description qu'impose un système pour un objet fixé. Les divers systèmes ne demandent pas à l'utilisateur la même entrée dans l'objet (nous verrons qu'ils ne présentent pas non plus à l'emploi les mêmes tâches, notamment de communication).

Voici quelques points de repère à l'appui de l'affirmation précédente :

Le regard sur l'objet, un rectangle par exemple, diffère si l'on use d'un système S1 (avoir une perception globale de la forme et connaître son appellation), S3 (connaître l'appellation et des paramètres de la forme), ou ...

1. Pour les logiciels à sémantique déclarative, nous proposerons l'analyse suivante : les instructions de production correspondent à des éléments primitifs pour décrire et/ou obtenir une description, éventuellement partielle, d'un objet complexe.

Pour un même type de système, en particulier pour S7, les instructions Ob_i peuvent exiger une description en rapport avec des propriétés de l'objet (perception locale orientée avec LOGO) ou une simple recopie (cas de PICTOR en mode dessin, où l'utilisateur peut procéder par tracé direct à l'écran avec le crayon optique). Avec certains systèmes, les instructions sont des appels de fichiers programmes ou données (S1-procédures prédéfinies en LOGO). Avec d'autres, elles consistent en données à fournir à des algorithmes désignés ou non par l'utilisateur (S2 à S6 - FRISE, MOSAÏQUE, SIMUL). Enfin, l'entrée dans un algorithme général d'interprétation peut être permis (S7, S8 - LOGO, PICTOR).

La vision et la description de l'objet sera plus ou moins libre, plus ou moins complexe suivant les instructions, c'est-à-dire le langage de commande ou de programmation disponible.

Nous avancerons qu'en type B, c'est-à-dire avec une communication II-I, la production a essentiellement lieu par donnée de paramètres à un unique algorithme spécifique: il n'est pas possible de faire afficher plusieurs objets en même temps à l'écran. La vision est fortement dirigée par les paramètres.

En type C, il peut exister plusieurs algorithmes dissociés (procédures) et un algorithme général d'interprétation, avec en sous-type II+ un mode différé. La vision est dirigée, mais dans une moindre mesure.

Suivant le cas, la description que fera l'enfant de l'objet pourra aller jusqu'à une explication à la machine des tâches requises pour la production (jusqu'à la donnée d'un algorithme complet pour l'objet en cause, - pour mémoire, dans le cas de logiciels à sémantique déclarative, de déclarations suffisantes).

Nous discuterons ultérieurement la réalité de l'emploi d'un langage, de la production d'un algorithme et d'un programme par l'enfant, dans le cas des langages de commande comme de programmation.

Nous nous contenterons ici d'attester sur la base des réflexions précédentes que la problématique de la primitivité des systèmes d'instructions des logiciels à communication II doit se placer dans une perspective didactique élargie.

Pour juger d'un système d'instructions de production d'un logiciel à communication II, il faut s'intéresser à:

- l'objet à produire.
- la description imposée au sujet par le couple «objet O, système S_k», du point de vue mathématique et du point de vue de la tâche de communication.

- l'état de connaissance du sujet eu égard à l'objet mathématique en cause et au système de communication proposé.
- l'objectif attribué à l'emploi du logiciel.
- l'utilisateur concerné (maître ou enfant).

L'examen de la primitivité des systèmes d'instructions de production d'un logiciel se replace ainsi dans celui des enjeux du logiciel, dont ce chapitre réalise une introduction.

LES TACHES DE COMMUNICATION

Quel que soit le type de communication (I ou II) proposée par le logiciel, au cours d'une activité, l'enfant fait face à une double tâche : une tâche relative à la communication et une tâche relative au contenu. Nous évoquerons ici les tâches spécifiques de communication. Nous nous interrogerons sur les problèmes posés, et tenterons d'évaluer la possibilité d'un traitement séparé de la tâche de communication. Nous ne prendrons pas en compte les tâches dues aux éventuels dysfonctionnements du logiciel, ce qui ne signifie pas que nous en minimisons l'importance.

1- TACHES DE COMMUNICATION DE TYPE I ; APPRENTISSAGE

Nous nous intéressons ici à la communication de type I : les logiciels concernés sont essentiellement ceux de type A, voire ceux de type B présentant des caractéristiques I.

1a Position du problème

Les logiciels à communication I impliquent des tâches de communication non directement apparentes. Celles-ci sont rarement reconnues par l'enseignant : l'apprentissage des règles de communication I fait le plus souvent l'objet de contrats implicites intervenant et évoluant en cours d'activité.

1b Définition des tâches de communication I

L'enfant rencontre une tâche de lecture de l'écran, non assimilable à celle de lecture d'un document graphique.

Nous évoquerons en premier lieu la lecture de l'écran au niveau de la forme. La taille de l'écran est limitée, son format est distinct des formats graphiques usuels. L'affichage de couleurs, d'inversion vidéo, de graphismes animés, d'émissions sonores est fréquent. Ainsi, la lecture doit changer, - n'oublions pas non plus que les écrans d'un logiciel ne se feuilletent pas comme les pages d'un livre -, s'adapter à de nouveaux indices. Ceux-ci sont en plus grand nombre. Ne peut-il être question d'une surabondance d'indices? Sont-ils pertinents, suffisants pour l'activité?

Nous continuerons par l'étude de la lecture de l'écran pour son contenu, c'est-à-dire par l'analyse à proprement parler des tâches relatives aux règles de communication.

Les règles de communication sont transmises par des instructions données par la machine. Elles concernent :

- L'activité elle-même (règles sur le contenu à communiquer, par exemple «dénombrer les carreaux d'une grille rectangulaire»).

Compte-tenu de la place disponible sur l'écran, la formulation de ces instructions peut obéir à une syntaxe et à une sémantique très pauvres, voire être sybilline. L'enfant doit alors s'employer à éclaircir ce qui peut apparaître comme un contrat implicite sur la définition de l'activité. A l'inverse, si l'écran est très chargé, comment l'enfant fera-t-il sa prise d'indices?

Quel équilibre satisfaisant pour l'activité peut-on trouver? Est-il nécessaire que tout figure à l'écran? Quels seraient les enjeux d'une dévolution de l'activité par l'environnement du logiciel (environnement graphique ou humain)?

Signalons que cette dernière option, retenue pour la conception des logiciels SIMUL et RECT, a donné des résultats appréciables, en particulier pour l'intégration du travail avec logiciel à l'ensemble des séquences d'enseignement, comme pour la négociation de l'activité avec l'ensemble du groupe (thèse de D. Butten "Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire").

- La forme que doivent revêtir les entrées utilisateur¹ (règles de transmission du contenu, par exemple «écriture obligatoirement de forme axb»).

Nous relevons une première difficulté pour l'élève : les écrans peuvent être mixtes (instructions concernant le contenu, instructions concernant sa forme).

Les règles de transmission du contenu par l'utilisateur, auxquelles doivent se conformer les entrées utilisateur, sont de deux types. Nous distinguons les «règles matérielles» concernant l'outil de transmission et les «règles logicielles» définissant la forme du contenu, ce dernier étant déterminé par ailleurs par les règles attachées à l'activité elle-même.

1. Rappelons brièvement que les entrées utilisateur ne sont pas dans les logiciels de communication I des instructions, sauf, le cas échéant pour le parcours du logiciel.

⇨ LES REGLES LOGICIELLES

L'enfant a trouvé une réponse au problème posé. Celle-ci est, rappelons-le, une construction personnelle n'utilisant pas de moyens logiciels. Il s'affronte alors, le cas échéant, à une tâche d'ajustement de sa production à une forme admise par le logiciel : il s'agit pour lui de fournir une représentation particulière de sa réponse.

Il pourra notamment être question d'un codage, d'une réécriture de la production primitive. Si le périphérique d'entrée est le clavier, on peut rencontrer des exigences sur le nombre de caractères, la présence de signes particuliers entre caractères, la nécessité d'un caractère de fin de réponse. La formulation de la réponse suit une normalisation pour satisfaire à un format logiciel (morphologie et syntaxe).

Ce formatage de réponse fait souvent l'objet d'un contrat implicite. Citons quelques exemples :

- Les règles relatives au caractère de fin de réponse sont tacites. S'agissant généralement de la touche de validation, son appui peut s'avérer indispensable ou non suivant que l'entrée peut se faire avec un nombre variable de caractères ou non. Dans un même logiciel, les deux cas de figures peuvent se présenter (entrées à nombre constant de caractères, entrées à nombre variable de caractères). A un endroit où le caractère de fin de réponse pourrait être omis (entrée à nombre constant de caractères), le logiciel peut l'imposer (pour respecter la cohérence avec d'autres entrées) ou non (pour accroître l'interactivité).
- Les instructions concernant le codage de réponses peuvent être données partiellement, selon un langage implicite. On rencontre fréquemment des questions du type ".....? (O/N)" ou ".....? (Oui/Non)", sous-entendant qu'il convient de taper O pour une réponse positive, et N dans le cas contraire.
- Il en est de même pour la précision du format que doivent adopter certaines entrées : ainsi, au chargement de logiciels peut s'afficher un message du type "Donner l'heure hh.mm.ss", qui invite à taper par exemple 15.05.00 (suivi ou non de la touche de validation) pour quinze heures, cinq minutes.
- Enfin, nous mentionnerons ici les instructions de branchement permettant à l'utilisateur d'accéder à la possibilité d'entrer une réponse, de parcourir, voire de sortir du logiciel. Elles sont rarement clairement exposées. Leur rappel figure parfois au bas de l'écran sous forme symbolique (CNTA, RAZ, -, ...). Dans certains cas, rien n'est indiqué, ni par le logiciel, ni sur sa documentation, alors que la réinitialisation du micro-ordinateur peut s'avérer être la seule façon de reprendre l'activité! Dans tous les cas, il est peu fréquent que l'enseignant étudie avec sa classe le moyen de parcourir le logiciel.

Le formatage des réponses, comme les règles de parcours du logiciel ne sont que peu normalisés sur un même logiciel ou d'un logiciel à l'autre. On assiste à des tentatives en ce sens qui ne remédient qu'imparfaitement à l'absence de reconnaissance par l'enseignant des tâches dues à la communication I.

Ainsi, pour certains aspects du parcours du logiciel, les produits diffusés par l'éditeur CEDIC/NATHAN tentent de conserver les mêmes instructions (ACC pour avoir des informations en début d'activité, RAZ pour quitter l'activité, INS pour adapter les données du logiciel à sa classe). En réponse au cahier des charges Education Nationale, les logiciels MEN/CNDP réclament systématiquement la touche → pour tourner les pages écran. Il nous paraît souhaitable dans le but d'une réduction des tâches de communication I que de telles normalisations interviennent, au moins pour le parcours du logiciel.

Dans certains cas, la tâche de communication est une tâche supplémentaire, non pertinente pour l'activité mathématique. On peut voir là une raison de sa non-reconnaissance par le maître : l'ajustement au format logiciel intervient après l'activité mathématique, lui est extérieur. Nous soulignons aussi qu'il y a risque d'appauvrissement, de réduction des moyens d'expression de l'élève. Les «langages-auteur» constituent-ils réellement des solutions par les libertés qu'ils offrent en matière d'analyse des réponses?

Dans d'autres cas, cette tâche fait partie intégrante de l'activité mathématique.

Ainsi, le logiciel TACHE cité dans la thèse de D. Buttien demande en accord avec des résultats didactiques relatifs à la multiplication, la donnée du nombre d'éléments d'une grille sous forme d'écriture multiplicative «axb».

L'outil ordinateur exige et contrôle alors mieux que quiconque le respect d'une consigne sur la forme de réponse, et ceci au profit de l'apprentissage mathématique. En conséquence, il serait intéressant de recenser des problèmes porteurs d'une exigence didactique réelle sur la formulation de la réponse et de construire des logiciels qui y correspondent.

⇒ LES REGLES MATERIELLES

La transmission à proprement parler d'une réponse formatée au logiciel qui la réclame met en cause une autre série de tâches : discerner ou choisir un périphérique d'entrée parmi plusieurs, connaître son usage.

Nous avons noté que le périphérique d'entrée est rarement signalé explicitement. En outre, il peut être différent suivant l'endroit du logiciel ou

l'on se trouve. D'autre part, son maniement fait exceptionnellement l'objet d'un apprentissage. Or, la maîtrise d'un clavier ou d'un crayon optique n'est pas immédiate. On peut s'étonner de l'oubli de l'importance de ces apprentissages en secteur éducatif, quand on constate que le domaine commercial y est sensible : ainsi, la documentation d'un appareil comme le «Macintosh» (APPLE) commence par une initiation de l'utilisateur à l'emploi du périphérique d'entrée qu'est la «souris».

1c Apprentissage de la communication de type I

Les tâches mises à jour justifient-elles un apprentissage? Comment peut-il être réalisé?

L'observation d'enfants ou d'adultes confrontés à la communication de type I montre leurs difficultés à résoudre les tâches en cause. Il est significatif que les réactions d'enseignants en train d'analyser un logiciel portent essentiellement, — et sans qu'ils le caractérisent comme tel dans un premier temps —, sur l'aspect communication du logiciel et non sur le contenu mathématique. Il est possible de voir là une manifestation des problèmes qu'ils rencontrent. La réalisation de l'activité mathématique peut se trouver gravement perturbée, la perplexité de l'utilisateur pouvant aboutir à son inactivité. Le contexte psychologique est aussi déterminant. L'utilisateur néophyte en matière d'usage d'un micro-ordinateur peut percevoir ses difficultés comme autant de preuves de son incompetence : ainsi, lorsqu'une option n'est pas disponible sur un logiciel (sortie de l'activité par exemple), il lui arrive d'être persuadé de son incapacité à la faire fonctionner.

Sur la base du descriptif des logiciels à communication I et du précédent examen des tâches de communication I, nous résumerons maintenant les explications que nous voyons aux difficultés de l'élève.

Ces logiciels n'imposent pas "naturellement" comme ceux à communication II une présentation d'informations préalables à leur emploi. L'enfant peut être mis en demeure de les utiliser sans préliminaire. Les règles de communication peuvent ne pas être comprises, d'autant qu'elles ne sont pas clairement édictées. Or, nous avons souligné qu'elles font précisément l'objet de contrats implicites établis en cours d'activité et évoluant avec celle-ci.

En outre, comme nous le verrons ultérieurement, les logiciels à mode réponse incitent l'enfant à ne pas prendre d'initiative, à attendre des directives d'une «machine-maître». En pleine contradiction, celles portant sur la communication sont le plus souvent au moins partiellement inexistantes : l'élève espère alors des ordres qu'il ne reçoit pas. Dans le cas contraire, si des indications relatives à la communication figurent à l'écran, elles ne sont pas reconnues, sont délaissées : nous avons observé que la lecture des écrans est

sélective et se fait surtout pour le contenu de l'activité. Une raison peut en être la centration sur la réponse au problème mathématique imposée par ces logiciels, ou encore la surabondance d'informations avec mixité sur le même écran, ou encore l'hermétisme d'instructions non explicitées (CNTA, RAZ, \rightarrow , ...), ou encore la croyance inconsciente en une incapacité à maîtriser totalement l'emploi d'un micro-ordinateur.

Nous soutiendrons qu'un apprentissage est nécessaire. Il ne peut intervenir qu'au niveau de la lecture de l'écran, du repérage et de l'emploi des principaux périphériques d'entrée, de la reconnaissance de la nécessité d'un formatage des entrées, de la détermination des instructions de parcours licites.

Cet apprentissage consistera surtout en une préparation à des éventualités en matière de lecture-écriture, autorisant une lecture des logiciels donnant du sens aux implicites, induisant quasiment des actes réflexes en cours d'utilisation. Il peut s'effectuer au travers de divers logiciels.

C'est le parti que nous avons pris au cours de formations d'instituteurs ou des produits informatiques de diverses disciplines étaient proposés aux stagiaires avec un objectif de maîtrise relative de la communication. Outre les gains dans ce domaine et leurs conséquences dans l'abord facilité des logiciels de mathématiques pour leur analyse en termes didactiques, nous mentionnerons l'intérêt d'une telle démarche comme préparation à la présentation de la communication de logiciels à communication plus complexe, tout spécialement de langages (logiciels à communication II). Les contraintes syntaxiques et sémantiques des logiciels à communication II sont alors perçues comme règles que l'on peut exposer et qui retiennent l'attention, puisque l'utilisateur a déjà pris conscience sur des objets plus simples de la nécessité d'une discipline de communication dans l'usage d'un logiciel. Par ailleurs, plus directement en liaison avec une préoccupation de didactique des mathématiques et en toute cohérence avec notre propos qui ne souhaite particulariser aucun logiciel avant une réelle analyse, cette méthode fait effectivement aborder en particulier LOGO aux utilisateurs comme un logiciel non différencié a priori des autres.

Un tel apprentissage permet d'envisager pour un problème donné une réduction de la tâche de communication I, que pourrait conforter une normalisation des instructions de parcours. Il ne faut pas pour autant oublier que celui-ci ne saurait être universellement valide. L'enseignant qui propose à des élèves un logiciel à communication I doit rester conscient de la spécificité de la tâche de communication attachée à ce logiciel. Il peut s'employer à l'explicitier, le cas échéant hors du logiciel, avec la classe.

2- TACHES DE COMMUNICATION DE TYPE II ; APPRENTISSAGE

Les logiciels concernés sont ceux de type B et C, mais aussi ceux de type A qui proposent pour la production d'un objet intermédiaire une communication de type II (sous-types I I-II⁻ et I I-II⁺ du type A).

L'étude que nous allons mener reprend certaines interrogations concernant la primitivité des systèmes d'instructions de production. Elle s'intéresse également aux autres éléments de la communication de type II, en particulier aux instructions de parcours que l'on rencontre en sous-type II-I.

Notons que les tâches matérielles de communication (choix et usage du périphérique d'entrée) font encore ici l'objet d'un contrat implicite sur lequel nous ne reviendrons pas.

2a Position du problème

Nous avons signalé que les tâches de communication I font l'objet d'un contrat implicite, rarement perçu par le maître. A l'inverse, les tâches de communication II, notamment celles relatives aux instructions de production, font généralement l'objet d'un contrat explicite reconnu par le maître.

Il convient toutefois de nuancer.

L'affirmation précédente s'avère essentiellement exacte en sous-type II⁺, voire II⁻ (par exemple, on apprend la programmation ou plutôt l'usage du logiciel LOGO). En sous-type II-I, certains logiciels (par exemple, MOSAÏQUE ou FRISE) qui ne se présentent pas à l'évidence comme un ensemble d'instructions et/ou font intervenir d'autres instructions que celles de production, engagent dans une certaine mesure un contrat implicite analogue à celui inhérent à la communication I.

Par ailleurs, lorsque la tâche de communication est reconnue, elle n'est pas nécessairement distinguée de celle portant sur l'objet d'enseignement mathématique.

Nous voyons à ce fait plusieurs raisons possibles :

- l'analyse par l'enseignant de la tâche générale (contenu et communication) qui incombe à l'élève est incomplète ou insuffisamment structurée.

- l'enseignant adhère à des thèses concernant la fonction de certains environnements informatiques pour l'apprentissage ("objet pour penser avec") et la forme d'apprentissage qu'ils autorisent ("apprentissage autonome"). Il est fait ici référence aux thèses de S. Papert et à l'environnement LOGO.

L'ensemble de ces considérations nous paraît justifier l'étude qui suit. Notons d'ailleurs que les problèmes posés par la communication de type II pour l'analyse de la tâche et la construction de séquences d'enseignement appropriées, - en particulier par le sous-type II⁺, où l'on trouve, outre des

systèmes d'instructions de production plus ou moins sophistiqués, des possibilités organisationnelles (itération, conditionnelle, récursion, ...), sont reconnus et étudiés au titre de recherches en didactique de l'informatique par des chercheurs dont N. Balacheff, J. Rogalski, A. Rouchier, R. Samurçay. Nous aimerions attirer l'attention sur l'intérêt de leurs travaux en y faisant référence.

2b Définition des tâches de communication II

Nous présenterons un panorama général des tâches que peut impliquer un logiciel à communication II. Nous préciserons, le cas échéant, au cours de l'exposé ce qui concerne un sous-type particulier.

Nous ne pensons pas devoir nous limiter, comme en communication I, à une description de tâches strictement logicielles. En effet, des aspects de la communication II (en particulier en sous-type II*) impliquent des tâches que nous qualifierons globalement de méthodologiques. Elles existent lorsque l'on traite certains problèmes avec certains logiciels. Si elles ne sont pas à proprement parler logicielles, elles ne sont pas non plus attachables au seul contenu mathématique de l'activité proposée. La résolution de ces tâches est souvent indispensable à l'établissement de la communication II, ce qui explique notre choix de les aborder ici.

⇨ LES REGLES LOGICIELLES

Elles sont déterminées par les systèmes d'instructions de production, auxquels s'adjoignent éventuellement des instructions pour la mise en place de structures itératives, conditionnelles, ... (sous-type II*).

Nous distinguerons plusieurs sortes de tâches liées aux règles logicielles.

• Tâche de connaissance des instructions de production

Chaque instruction de production constitue un élément minimum de sens que l'utilisateur doit acquérir. Nous remarquons qu'ici la problématique est différente de celle de la communication de type I : il n'y a pas d'ajustement d'un signifié (construction personnelle d'ordre mathématique) à un signifiant, mais recherche d'un signifié pour chaque signifiant.

L'ensemble des instructions peut constituer un codage de signifiés (application bijective entre l'ensemble des instructions ou signifiants et l'ensemble des signifiés). Il peut aussi en être autrement : certains logiciels offrent des instructions dont le sens varie avec la situation, c'est-à-dire qu'il existe des signifiants identiques pour des signifiés distincts.

Par exemple, pour les logiciels RECT ou SIMUL, on note que le sens des instructions «barre d'espace», «L», «C», change suivant

les différents niveaux de ces logiciels où elles sont valides.

L'instruction «barre d'espace» produit :

- au niveau I, l'affichage du reste de la ligne (respectivement colonne) en cours et, s'il ne s'agit pas de la dernière ligne (respectivement colonne), l'affichage du point de la ligne (respectivement colonne) suivante de la grille.

- au niveau Z, une stratégie d'affichage (à savoir la répétition de celle qui vient d'être employée).

L'instruction «L» (respectivement «C») produit :

- au niveau X, suite à une instruction «P» ou «A» au niveau Y, l'affichage de la grille ligne par ligne (respectivement colonne par colonne).

- au niveau I, suite à une instruction «M» au niveau Y, la modification du type d'affichage, c'est-à-dire la commutation - sans poursuite automatique de l'affichage - d'un affichage colonne par colonne en affichage ligne par ligne (respectivement la commutation d'un affichage ligne par ligne en affichage colonne par colonne). Au regard de l'utilisateur, dans ce cas, les instructions «L» ou «C» sont des commutateurs, alors que dans le contexte précédent, elles génèrent des objets à l'écran.

Dès lors, la tâche de l'élève ne s'en trouve-t-elle pas accrue (ambiguïté, particularité)?

Nous émettrons à propos de la tâche d'acquisition de signifiés une hypothèse :

L'importance, ce qui ne veut pas nécessairement dire la difficulté, de la tâche dépend :

- du nombre d'instructions mis à disposition de l'utilisateur pour la production de l'objet en cause dans le problème.

- de l'indice du système d'instructions, les systèmes à indice élevé (présents surtout dans les logiciels de sous-types II+ ou II-) étant susceptibles d'engendrer une tâche plus importante.

Importance est ici à prendre au sens numérique, quantitatif.

Nous rappelons au sujet de la première partie de cette hypothèse que les instructions ne figurent pas toujours à l'écran et peuvent donc être fournies à discrétion par l'enseignant.

Pour étayer la dernière partie de l'hypothèse, nous redirons que les systèmes à indices élevés obligent généralement à une plus grande description de l'objet en termes de sous-objets et opérations : par conséquent, l'utilisateur peut être amené à devoir maîtriser le sens d'un plus grand nombre d'instructions. Ceci est évidemment relatif et ne sera mesurable qu'avec la donnée explicite d'un problème à traiter avec un ou plusieurs systèmes d'instructions.

La tâche d'acquisition de signifiés est généralement explicite en sous-types II- et II+.

Notons que, pour la plupart de ces logiciels, les instructions sont décrites sur un document d'accompagnement ; seuls figurent parfois à l'écran des pictogrammes représentatifs des instructions (PICTOR) ou des rappels. La présence des instructions sur document d'accompagnement fonde-t-elle la reconnaissance de la tâche par l'enseignant? S'agit-il d'un avantage pour l'activité mathématique (écran moins chargé, possibilité pour l'utilisateur de considérer le logiciel comme un ensemble d'instructions extérieures à la machine, mis à sa disposition pour résoudre un problème)? L'existence d'un double matériel (document graphique et machine) entraîne-t-elle une tâche accrue? Certains logiciels permettent le traitement de situations très diverses (par exemple LOGO en sous-type II+) : leurs systèmes d'instructions ont un caractère de généralité que d'autres n'ont pas (par exemple SIMUL). Le critère de généralité doit-il seul être pris en compte pour décider de la mise à l'écran d'un minimum d'instructions?

Certains logiciels du sous-type II-I n'établissent pas toujours un contrat aussi explicite qu'en II- ou II+.

L'existence d'instructions et corrélativement d'une tâche qui lui est associée n'est pas évidente, surtout lorsque l'apparence du logiciel induit son affiliation au type I.

Le logiciel SIMUL se présente d'emblée comme un ensemble d'instructions, dont il impose la connaissance en affichant seulement à l'écran : "Donne une commande X (respectivement Y ou Z)". Le contrat est alors explicite et de type II- ou II+.

A l'inverse, FRISE, MOSAÏQUE ou DIVIS donnent lieu à un contrat plus implicite, proche parfois de celui de type I :

- les instructions de production de FRISE (désignation d'un groupe d'isométries, désignation de points de l'écran) apparaissent facilement à l'utilisateur non comme telles, mais plutôt comme des réponses qu'il doit apporter à une question qu'il lui arrive de formuler, une fois dépassé un premier stade de perplexité, de la façon suivante : "reproduire les pictogrammes qui figurent à l'écran" (il s'agit des pictogrammes correspondant aux sept types de frises). La raison de ce comportement peut résider en l'identification que réalise l'utilisateur du logiciel FRISE à un logiciel de type I. En effet, la méconnaissance du propos mathématique du logiciel (les sept types de frises du plan) conjuguée au contenu des écrans (coexistence dans une formulation semblable d'instructions pour la production et pour le repérage) autorisent un vécu de type I. Au demeurant, même une fois franchi le stade de reproduction des pictogrammes, le sujet peut n'admettre que difficilement qu'il détermine l'apparition de points à l'écran, que le problème qu'il traite n'est pas complètement fixé par le logiciel, qu'il apporte un complément de définition à la décision logicielle.

- DIVIS établit aussi un contrat de type I. La connaissance limitée qu'ont certains enseignants des résultats didactiques sur la division et l'aspect des écrans (écrans dont la présentation est analogue à celle de logiciels de type I) sont plus ici en cause que le contenu mathématique.

- Avec MOSAÏQUE, le contrat est de type II. L'utilisateur devance souvent l'instruction de repérage ("Donne le code ...") pour effectuer son entrée : dès lors, on peut penser que celle-ci a bien pour lui statut d'instruction de production. D'ailleurs, des enfants de CM2, négligeant le rappel possible des codes en cours d'activité, les avaient soigneusement notés. Le problème de construction posé de ce logiciel (choix de pavés qui, reproduits, formeront une mosaïque) nous semble expliquer le contrat de type II qui s'instaure malgré les formulations à l'écran par questions, caractéristiques du type I.

Les exemples précédents nous incitent à conclure pour le sous-type II-I :

La présence à l'écran d'instructions de production, le caractère mixte des écrans (instructions de divers genres), l'existence d'instructions de parcours conduisent à une perception nuancée des instructions de production comme telles, le logiciel pouvant être apparenté au type I de par sa présentation. La tâche relative au sens des instructions de production risque alors d'être ignorée de l'enseignant.

L'enfant ayant perçu le fonctionnement d'un logiciel en communication II peut être gêné par des interventions du logiciel qui coupent son activité : émission d'instructions par la machine (pas d'attente de l'instruction de repérage avec le logiciel MOSAÏQUE) ou demande à l'utilisateur d'une instruction de parcours branchement.

D'une manière générale, ne serait-il pas souhaitable que s'affirme la forme d'un logiciel en matière de communication, comme cela est le cas avec SIMUL ou RECT ou avec des logiciels du sous-type II- de plus en plus nombreux (CARACTOR, COLORPAINT, SCRIPTOR, ...)? Ne conviendrait-il pas également d'écrire des logiciels qui ne comporteraient pas d'instructions de parcours?

• Tâche de connaissance des règles de syntaxe

Cette tâche existe lorsqu'il est possible ou nécessaire d'enchaîner plusieurs instructions.

Elle peut s'avérer particulièrement importante en sous-type II+, où elle est susceptible de s'accompagner de tâches spécifiques à l'emploi de structures conditionnelles, itératives, ..., complétant les offres des systèmes d'instructions de production, ou même au recours à un mode différé.

Lors d'observations d'adultes ou d'enfants, nous avons pour notre part

identifié des difficultés didactiques en LOGO. Nous ne les mentionnerons pas ici, le sujet étant trop important pour mériter de simples remarques : nous préférons renvoyer le lecteur aux travaux de R. Samurçay en particulier. Nous attirerons toutefois l'attention sur le fait que la syntaxe de ces logiciels n'est pas analogue à celle des langages naturels et peut même s'avérer incohérente par rapport cette dernière.

Ainsi en LOGO, on peut être amené à demander la production d'un grand carré par une formulation du type CARRE "GRAND. Aujourd'hui, peu de logiciels obéissent à une syntaxe de type «sujet-verbe» : la norme est généralement «verbe-sujets».

Ce qui est une facilité pour le logiciel est une contrainte pour l'utilisateur, et réciproquement. Pour plus de performance dans ce domaine, il faudrait réaliser un système fort coûteux : ne serait-on pas alors ramené à des problèmes analogues à ceux rencontrés pour les analyses de réponse en communication I ?

En sous-type II-, avec des langages de commande, la tâche est moindre, l'utilisateur ayant souvent à gérer une arborescence limitée.

Elle est souvent inexistante en sous-type II-I où les instructions de parcours (type repérage) données par la machine règlent le problème : la structure peut être plate.

• Tâche de compréhension des messages émis par le logiciel par rapport au discours de l'utilisateur

Il s'agit de messages signalant des incorrections syntaxiques/sémantiques dans le discours de l'utilisateur ; ils ne portent pas sur un éventuel attendu conceptuel du logiciel.

Expliquons la formulation «syntaxique/sémantique» : L'analyse des incorrections peut s'effectuer de deux points de vue différents : celui de l'utilisateur, celui du logiciel. Ainsi, lorsqu'un utilisateur tape en Logo, AV10, en oubliant l'espace séparateur entre le V et le 1, il commet une erreur de syntaxe ; toutefois, le message envoyé par le logiciel est à contenu sémantique : "Ne sais comment faire AV10". Ceci explique la formulation «syntaxique/sémantique». Notons aussi que lorsque l'utilisateur tape REPETE 4 [AV 50 TD 50] pour obtenir un carré, il ne fait pas une erreur syntaxique, mais sémantique ; du point de vue du logiciel, il n'y a pas incorrection, et par conséquent, pas de message émis. On trouvera un examen des types d'erreur en programmation dans l'article de Patrick Mendelsohn "L'analyse psychologique des activités de programmation chez l'enfant" (Revue Enfance, n° 2/3, 1985, p. 213-221).

Nous distinguerons les messages qui indiquent le non-respect de codes ("Tu n'as pas respecté les codes" dans SIMUL), d'autres messages plus complexes.

La lecture des messages du premier genre s'opère généralement correctement. Elle est suivie d'un nouvel essai de l'utilisateur.

Les messages plus complexes renseignent sur le type et l'endroit de l'incorrection. C'est en particulier le cas des messages des logiciels qui sont des langages de programmation (sous-type II+).

Nous avons observé que, tant qu'une certaine connaissance des règles de communication régissant l'emploi du logiciel n'était pas assurée, tout se passait comme s'ils n'étaient pas lus. Plus exactement, assez rapidement, l'utilisateur néophyte ne cherche plus à percer leurs sens : il se contente d'enregistrer un signal synonyme de "cela ne va pas" qui l'amènera le plus souvent à vouloir reprendre l'activité à son début, sans tenter d'intégrer les informations reçues.

Nous avancerons qu'à chaque fois que le sens d'une instruction donnée par la machine n'est pas accessible à l'utilisateur en raison de l'insuffisance de ses savoirs, celle-ci n'est interprétée que comme un signal induisant un comportement circonstancié (étudement ou reprise complète de l'activité si possible, blocage sinon).

Cette remarque ne s'applique pas aux seuls messages d'incorrection : elle concerne tous les types d'instructions données par les logiciels à communication II, mais aussi à communication I.

Ainsi, D. Butten signale à propos du logiciel TACHE que l'élément de correction "Ton score est 10, celui de l'ordinateur est 9" n'est pas significatif pour les élèves, qui l'ignorent. Effectivement, la compréhension de ce message dépend directement de la maîtrise de l'activité mathématique proposée (dénombrement des points d'une grille rectangulaire, suite à l'affichage de points nécessaires et suffisants à sa détermination).

Comprendre, c'est-à-dire rendre opérationnels, les messages d'incorrection émis par des logiciels comme LOGO ou BASIC suppose une connaissance suffisante de ces produits informatiques.

En BASIC, le message "SN ERROR in 10" (erreur de syntaxe en ligne 10) ne sera efficace qu'à la condition qu'une lecture critique de la ligne 10 soit possible, ce qui suppose des savoirs en programmation BASIC.

Il en est de même en LOGO : nous ne pouvons suivre C. Berdonneau qui dans "Recueil de pratiques pédagogiques autour de LOGO" reconnaît à LOGO des messages "infiniment moins ésotériques" que ceux des autres langages.

Ainsi, "NE SAIT COMMENT FAIRE AV10" ne constitue pas un

message pertinent pour qui n'appréhende pas suffisamment la notion de procédure et l'importance du séparateur (espace) : en effet, ici, le logiciel avertit qu'il a tenté d'interpréter une procédure ayant pour nom AV10 (qui, bien sûr n'est pas définie), ceci en raison de l'absence du séparateur "espace" entre le V et le 1.

Dans le but d'une prise en compte des messages par l'élève, des enseignants institutionnalisent la tâche de leur lecture. Ceci ne résoud pas le défaut de connaissance de l'élève qui peut alors inventer un sens adapté à ses savoirs : ainsi le message LOGO "TABLEAU EST DÉJÀ DÉFINI" a été reçu par une fillette de CMI en cours d'acquisition de la notion de procédure, comme l'acquiescement de la machine à sa tentative de créer une procédure ayant pour nom TABLEAU (d'où sa perplexité ultérieure ...).

En conclusion, la tâche de lecture des messages émis par le logiciel relativement au discours utilisateur ne nous paraît résoluble que si les savoirs qu'ils impliquent sont déjà au moins pour partie acquis par l'élève. Dans ces conditions, que penser de la thèse d'«apprentissage autonome» attachée à LOGO, qui s'appuie pour partie sur l'aide apportée par lesdits messages ?

Par la citation de G. Bossuet dans "L'ordinateur à l'école" (page 60), nous marquerons l'expression de l'idéologie qui accompagne LOGO dans le domaine des messages du logiciel :

"Le système n'insulte jamais un enfant qui a oublié une règle de syntaxe, ne lui renvoie jamais des messages du genre «erreur n° x», mais il lui demande poliment soit ce qu'il voulait dire, soit des précisions. Le langage utilisé n'est donc *pas culpabilisant*."

• Tâches méthodologiques

Il s'agit de tâches qu'impliquent plus spécialement les logiciels de sous-type II+ (LOGO, ...).

Les larges possibilités qu'ils offrent en matière de communication peuvent exiger, suivant le problème donné, des méthodes d'analyse spécifiques pour la conception d'algorithmes adaptés à leur principe de fonctionnement logique.

L'importance de ces tâches diminue des logiciels de type II+ aux logiciels de type II-I, où le discours peut-être linéaire et corrélativement l'algorithme peu complexe.

Nous énumérerons ces tâches, tout en sachant qu'elles sont liées et déterminées par la logique du logiciel comme par la question mathématique à traiter :

- savoir préciser le problème à traiter (ce que souligne J. Rogalski dans son article "Alphabétisation informatique").
- savoir pratiquer une analyse de problème (analyse ascendante,

descendante, ...).

- savoir transformer une analyse en un algorithme performant pour le logiciel utilisé (techniques spécifiques à LOGO, BASIC, ...).

- savoir traduire un algorithme avec les instructions du logiciel utilisé (méthodes de programmation diverses ; représentation des données).

Ceci réclame des savoirs, des formes de pensée non innées. L'objet mathématique du problème n'est pas directement en cause, toutefois la résolution de celui-ci en dépend dans la mesure où elle passe par l'usage du logiciel.

2c Apprentissage de la communication de type II

Nous poserons des questions identiques à celles concernant la communication de type I : les tâches évoquées justifient-elles un apprentissage? Quand et comment peut-il être réalisé?

L'existence de systèmes descriptifs implique l'ajustement au format logiciel du traitement du problème posé, et non plus exclusivement comme en communication I de la réponse terminale. Cet ajustement intervient pendant le déroulement de l'activité et en fait donc partie intégrante. On peut y voir une nouvelle raison de la reconnaissance de la tâche de communication par l'enseignant. On peut aussi y trouver la nécessité d'un apprentissage de la communication : l'enfant ne pourra agir que s'il est en mesure d'user du logiciel.

Nous défendons pour notre part la nécessité d'un apprentissage spécifique préalable qui autorise la réflexion sur l'objet mathématique du problème de référence en réduisant la tâche de communication. Nous en examinerons plus loin les conditions et les limites.

C'est à partir de cette réflexion qu'avec D. Butlen, nous avons proposé le logiciel SIMUL aux élèves pour l'apprentissage des commandes mises en jeu par RECT.

On rencontre chez S. Papert et d'autres partisans du «projet LOGO», l'hypothèse d'un apprentissage "naturel" mêlant celui de la communication et de notions mathématiques. Le modèle d'apprentissage invoqué est "celui de l'enfant qui apprend à parler, processus qui se déroule sans enseignement délibéré et organisé" ("Jaillissement de l'esprit" S. Papert, page 19).

Ceci ne nous paraît fondé ni pour les mathématiques, ni pour la communication.

Notre examen des tâches de communication contredit une telle position. Nous ajouterons qu'au niveau sémantique, si les acquisitions relatives au sens peuvent précéder celles de type grammatical pour un langage naturel, il en va

autrement avec un langage artificiel : la structure syntaxique nous paraît première. D'autre part, nos observations ne nous engagent pas à croire qu'une activité puisse avoir un double objectif d'apprentissage mathématique et de communication.

A l'inverse de S. Papert et en accord avec A. Rouchier, nous concevons moins la machine comme un "objet pour penser avec" sans apprentissage préalable, que comme un système didactique dont une certaine maîtrise doit être assurée pour qu'il puisse remplir un rôle eu égard aux acquisitions mathématiques. Nous persistons ainsi à trouver la thèse de l'«apprentissage autonome» peu acceptable, dans la mesure où elle repose aussi sur l'apprentissage naturel à double objectif ici contesté.

⇒ A PROPOS DU DEGRE D'APPRENTISSAGE

• Quantité d'apprentissage

Quels sont les éléments que l'enfant doit acquérir pour mener une activité mathématique avec un logiciel?

Suivant le problème donné, il peut ne pas être nécessaire de maîtriser la totalité des objets de savoir relatifs à la communication II. Toutefois, quelles seraient les répercussions sur l'activité mathématique d'une plus grande appréhension de la communication ? Y-a-t-il un seuil ? Doit-on s'attacher à rendre disponibles des instructions qui le sont peu, car peu conformes à des pratiques extérieures dans d'autres domaines ?

Par exemple :

- La production d'un hexagone régulier avec LOGO n'impose pas la connaissance de l'instruction «REPETE», ni celle de la récursion et des arrêts conditionnels. Toutefois, si l'élève pouvait employer ces moyens, percevrait-il différemment un hexagone?

L'emploi de l'instruction «RECOULE» dans un contexte positif (dessiner en reculant) en LOGO est peu fréquent, car il ne correspond à aucun vécu extérieur.

- La connaissance des instructions de RECT était apparemment suffisante pour que les enfants affichent une grille : D. Butten signale à l'issue de la passation de SIMUL que "Il n'y pas eu d'erreur de frappe ou d'utilisation d'une commande d'un type à un moment inadéquat. Toutes les commandes ont été utilisées" (page 68). Toutefois, l'analyse des résultats de l'expérience avec le logiciel RECT, met en valeur des disparités dans le recours aux diverses instructions ; par exemple, l'instruction «M» est peu utilisée, alors qu'elle aurait permis de ralentir la machine. La disponibilité des instructions était-elle suffisante à une exploration sans contrainte des grilles, susceptible d'autoriser la production d'une écriture multiplicative?

L'emploi de «L» (affichage de la grille ligne par ligne) est majoritaire avec SIMUL, vraisemblablement en raison du respect que réalise cette instruction du sens usuel de lecture.

• Qualité de l'apprentissage

Il est bien connu que l'on peut communiquer dans une langue en ne maîtrisant pas complètement le discours pour son articulation ou le sens de tous ses éléments, ou encore en employant différents niveaux de langage.

Un logiciel n'autorise pas d'erreurs de syntaxe comme une langue naturelle. Néanmoins, il permet (essentiellement en sous-types II- et II+) l'expression suivant divers niveaux de langage.

Cela est-il admissible eu égard à un objectif d'apprentissage informatique que l'on pourrait avoir, outre celui d'ordre mathématique ? Quelle est l'influence sur l'apprentissage mathématique du niveau de langage dont l'utilisateur dispose ?

Ainsi, plusieurs expressions sont possibles pour un carré en LOGO. Par exemple :
 REPETE 4 [AVANCE 50 TD 90], ou encore AVANCE 50 TD 90 AVANCE 50 TD 90 AVANCE 50 TD 90 AVANCE 50 TD 90, ou encore AVANCE 30 AVANCE 20 TD 80 TD 10 AVANCE 50 TD 90 AVANCE 50 TD 90 AVANCE 50 TD 90, ou encore POLY 4 50 si POLY :N :L est une procédure paramétrée pour l'affichage d'un polygone régulier à N côtés de longueur L, ou encore ...

L'enfant peut aussi communiquer avec la machine et obtenir une réalisation correcte en utilisant des instructions auxquelles il affecte un sens mathématique peu correct.

Ainsi, l'instruction LOGO «TG 30» (rotation de trente degrés dans le sens trigonométrique) est interprétée par de jeunes enfants de cycle élémentaire en référence à l'instruction «AVANCE» : pour eux, la tortue effectue un déplacement, une translation, de trente petits pas correspondant au troisième côté du triangle bâti sur les positions de départ et d'arrivée. Le statut du nombre qui est argument d'une instruction de rotation (mesure d'angle) est assimilé à celui du nombre qui est argument d'une instruction de translation (mesure de longueur).

Peut-on se contenter d'une connaissance de la communication reposant sur des représentations mathématiques inexactes ? La communication n'en est pas affectée, mais qu'en est-il de la construction des concepts mathématiques ?

• Disponibilité de l'apprentissage

Demande-t-on aux savoirs d'être disponibles à un moment donné pour un ensemble d'activités mathématiques précis, ou au contraire en eux-mêmes?

Des enfants de CM2 étaient capables de produire des programmes structurés pour la réalisation à l'écran d'assemblages de carrés. Par contre, lorsqu'il leur a fallu aborder le problème

mathématique complexe du dessin d'une rosace non connexe, la tâche mathématique (analyse de la rosace en terme de rotation, détermination du centre et de l'angle) a provoqué un retour à un mode de communication linéaire.

En outre, un parallèle peut être fait avec la maîtrise de la langue : la liberté de pensée sur l'objet mathématique du problème passe-t-elle par la mémorisation des instructions du logiciel ? Une exigence de mémorisation peut-elle être soutenue si le logiciel ne présente pas un caractère de généralité (par exemple, cas du logiciel SIMUL) ?

⇒ A PROPOS DES CONDITIONS ET LIMITES DE L'APPRENTISSAGE

Les instructions de production sont disponibles avant le recours au logiciel pour une activité mathématique précise. On peut donc penser que l'apprentissage de la communication peut précéder sans problème majeur et à son bénéfice ladite activité. Qu'en est-il ?

• Accessibilité de l'apprentissage

Si le logiciel et/ou l'activité réclament un mode de pensée particulier (pensée logique de type algorithmique, pensée structurée, pensée récursive), celui-ci est-il accessible à tout âge ?

D'autre part, nous avons écrit qu'il était envisageable de faciliter la résolution de problèmes de syntaxe élémentaire en habituant l'utilisateur aux exigences générales d'un logiciel : on peut lui soumettre divers logiciels (notamment de sous-type II-) qui lui apprendront à gérer un certain nombre de règles concernant les périphériques et les formats d'entrée, un transfert s'effectuant sur des logiciels plus sophistiqués (en particulier de type II+, LOGO par exemple).

Toutefois, l'acquisition de savoirs plus précisément attachés au logiciel que l'on veut utiliser fait intervenir un contexte mathématique. En effet, les logiciels qui nous intéressent impliquent, ne serait-ce que par le sens de leurs instructions, des notions mathématiques. L'apprentissage de ce sens, ou celui de notions spécifiques de communication ne peut s'opérer dans le vide : l'économie d'activités ayant un contenu mathématique n'est pas vraisemblable. Ainsi, l'apprentissage de la communication ne peut être que contextualisé. Dès lors, est-il possible si un certain contenu mathématique n'est pas acquis ?

Par exemple, en LOGO, nous ne pensons pas qu'un enfant puisse utilement aborder la notion de procédure à travers la réalisation d'un graphisme s'il ne maîtrise pas les instructions graphiques pour leur sens, ou dans le cas contraire, si le graphisme proposé met en cause une activité mathématique trop complexe. En effet, la tâche d'analyse mathématique serait dans les deux cas trop prégnante.

Par ailleurs, si les connaissances mathématiques de l'enfant sont telles que des activités relatives à la communication ne peuvent lui être confiées que dans des domaines mathématiques très restreints, quel sera le champ de validité des apprentissages logistiques ?

Nous avons déjà évoqué l'absence de validité de la méthode de programmation structurée pour une rosace non connexe, compte-tenu de la difficulté de la tâche mathématique pour des enfants de CM2.

Nous soulignons ici que, comme c'est le cas pour des acquisitions dans le domaine numérique, les acquisitions logistiques ne sont pas uniformes et indépendantes du champ où on les exerce : il n'y a pas de transfert automatique d'un domaine d'application à un autre.

Pour aller plus loin, certains logiciels (en particulier de sous-type II+) font intervenir des concepts spécifiques de communication : notion de procédure, de paramètre, de variable, d'itération, de récursion. Nous pensons que l'apprentissage de ces concepts doit intervenir dans des contextes variés susceptibles de leur donner un sens, comme cela est souhaitable pour l'apprentissage de concepts de type mathématique¹. Pourra-t-on construire des séquences didactiques appropriées et diversifiées, si par ailleurs le domaine mathématique dominé par l'enfant est limité ?

• Obstacles à l'apprentissage liés aux savoirs du sujet

Les connaissances générales, mathématiques, ou même en matière de communication du sujet peuvent constituer des obstacles à l'apprentissage.

Ainsi, par exemple :

- Des acquisitions en géométrie euclidienne peuvent contrarier des apprentissages en LOGO, ce logiciel faisant référence à une géométrie différentielle.
- La pratique d'une communication de type I peut engendrer des comportements peu propices à l'abord d'une communication de type II.
- La connaissance en sous-type II+ d'une approche spécifique (par exemple, algorithmique procédurale avec BASIC) peut en handicaper une autre (par exemple, communication applicative avec LISP ou LOGO).

1. Des recherches en didactique de l'informatique traitent cette question : par exemple, R. Samurçay construit des situations d'apprentissage relatives à la notion de procédure paramétrée en LOGO.

• Apprentissage de la communication et apprentissage mathématique

L'apprentissage de la communication proposée par des logiciels utilisables à des fins d'apprentissage mathématique est contextualisé dans le domaine mathématique. En outre, nous souhaitons qu'il précède dans la mesure du possible l'apprentissage mathématique à proprement parler.

Dans ces conditions, quelles séquences spécifiques ayant pour objectif un apprentissage logistique peut-on mettre au point ? Quelles sont les conséquences sur l'activité mathématique projetée ? Est-il possible d'interférer au minimum ? L'étude préalable de la communication (notamment, des systèmes d'instruction de production) ne risque-t-elle pas de vider le logiciel de son intérêt didactique pour les mathématiques ?

Une hypothèse peut être retenue en réponse aux questions précédentes :

La possibilité de déconnecter l'apprentissage de la communication de l'activité mathématique envisagée dépend de ladite activité, du logiciel et de l'état de connaissance (mathématique et communication) du sujet.

Ainsi, les logiciels de type B (communication de sous-type II-I) offriront moins de liberté pour un apprentissage isolé de la communication que les logiciels de type C (communication de sous-types II+ et II-). En effet, les premiers définissent partiellement le contenu de l'activité pour ne laisser éventuellement qu'un complément de définition possible sur la démarche. Il peut donc être délicat, voire improbable de concevoir une autre question que celle incluse dans le logiciel ; par contre, cela est envisageable avec un logiciel de type C qui ne définit pas de contenu d'activité.

L'apprentissage des instructions de SIMUL (type B, communication II-I) s'est opéré directement sur l'objet «grille de points» dont l'analyse est le but même du logiciel.

L'apprentissage des instructions de RECT (type A, communication I-II+ pour la réalisation de l'objet intermédiaire «grille de points») a été menée avec SIMUL, et par conséquent très près de la question posée par RECT, à savoir la production d'un dénombrement multiplicatif pour une grille de points.

Pourrait-on concevoir un apprentissage du sens de l'instruction «E» de TACHE (type A, communication I-II+) qui ne soit pas la résolution du problème posé par le logiciel ? Cela semble peu vraisemblable.

Avec les logiciels de type C, les connaissances (mathématique et communication) de l'enfant sont des facteurs non négligeables :

- Avec des enfants du cycle élémentaire, l'étude de formes géométriques avec LOGO et l'acquisition du sens d'instructions graphiques de ce logiciel risquent d'être difficilement dissociables.

- A l'inverse, l'étude d'une rosace avec LOGO pour des élèves du cycle moyen peut succéder à un apprentissage méthodologique

relatif à la communication (méthode de programmation structurée), dont ... nous avons déjà souligné les limites. Celui-ci ne fera pas nécessairement intervenir les notions mathématiques sous-jacentes au problème de la rosace (il est possible par exemple de proposer des travaux autour d'assemblages non réguliers ou présentant des régularités d'un autre genre).

⇒ QUELQUES PROPOSITIONS POUR L'APPRENTISSAGE

On peut chercher à concevoir des situations ne présentant pas pour le sujet de difficultés mathématiques majeures.

En formation d'adultes, la réalisation avec LOGO d'un carré, puis d'un rectangle, puis d'un triangle équilatéral répond aux objectifs de maîtrise du sens des primitives de déplacement (activités : carré et rectangle), puis de compréhension du repérage local pour la détermination des angles (activité : triangle équilatéral), ceci sans intervention d'une tâche mathématique importante pour le sujet. Il est aussi possible d'aborder la méthode de programmation structurée en LOGO en fournissant à l'utilisateur un lot de procédures prédéfinies par l'enseignant pour la production d'un graphisme. L'analyse des listings servira lors d'une institutionnalisation.

Lorsque cela est permis par le logiciel, on peut renommer les instructions ou en adjoindre de nouvelles au lot existant pour constituer un corpus dont l'emploi sera moins complexe ou plus adapté aux objectifs poursuivis.

Ceci est licite en LOGO. On peut citer le complément d'instructions constitué par un lot de procédures prédéfinies dans le contexte évoqué ci-dessus. On peut aussi mentionner la décision d'une enseignante de CM2 qui, suite aux difficultés des élèves à appréhender correctement le sens des primitives de rotation «TD» et «TG», les a remplacées temporairement par «HD» et «HG», c'est-à-dire heure droite et heure gauche prenant pour arguments des nième de tour, en référence à la rotation des aiguilles d'une horloge.

On peut tenter de remplacer les périphériques usuels d'entrée-sortie, c'est-à-dire de modifier l'environnement de communication, pour diminuer (ou tout au moins transformer) les tâches matérielles et/ou logicielles de communication.

Le "Promobile Jeulin" (tortue de sol) apparaît comme une tentative en ce sens relativement au logiciel LOGO. Un lecteur de cartes perforées selon des primitives LOGO peut remplacer le clavier, tandis que la tortue demi-sphérique tient lieu de périphérique de sortie. Divers signaux sonores sont émis par cette configuration. Un examen comparatif de LOGO-écran et

LOGO-tortue de sol peut être mené sur le plan de l'apprentissage de la communication et de l'apprentissage mathématique. Signalons que des élèves de cycle moyen ont eu recours à la tortue de sol pour mieux construire le sens des primitives de rotation, par visualisation des mouvements du robot-tortue.

Le crayon optique ou la souris sont aussi des périphériques d'entrée censés modifier en l'allégeant la tâche de communication (pas de manipulation de clavier ; adressage direct, ou à partir du plan de la table, de l'écran).

On peut tenter de définir des situations n'employant pas de matériel informatique, mais impliquant des savoirs qui pourront être réinvestis.

Evoquons quelques possibilités :

- activités visant à un apprentissage de type algorithmique.
- activités de codage-décodage.
- jeux de rôle en LOGO avec les élèves.
- plus généralement, activités de communication au sens courant du terme.

2d Conclusion relative aux tâches de communication de type II

La difficulté de la tâche de communication II ne s'exprime pas exclusivement par la complexité des objets de savoir intrinsèques à la communication II, mais aussi par l'état de connaissance du sujet par rapport à la communication en général, aux mathématiques, au problème à traiter.

L'apprentissage de la communication de type II se place dans une dialectique apprentissage mathématique, apprentissage de la communication.

3- CONCLUSION RELATIVE AUX TACHES DE COMMUNICATION

L'usage de micro-ordinateurs dans l'enseignement des mathématiques s'assortit de l'apparition de nouvelles tâches, des tâches de communication. L'activité mathématique peut dépendre de leur résolution. Elles nous semblent suffisamment complexes pour justifier un apprentissage spécifique préalable, dans la mesure du possible.

C'est pourquoi, nous souhaitons attirer l'attention sur leur existence, d'autant qu'elles risquent de faire l'objet de contrats implicites (notamment, en communication de type I).

La nature des tâches et les conditions d'apprentissage de la communication varient d'un type de communication à l'autre.

Par conséquent, nous terminerons en nous interrogeant sur la tâche globale impliquée par les logiciels de sous-type I-II' ou I-II' du type A,

ceux-ci faisant appel à une communication de type I pour fournir la réponse au problème posé et à une communication de type II pour construire un objet intermédiaire. Nous différerons au chapitre concernant la tâche relative au contenu de l'activité, l'examen des conséquences pour l'apprentissage mathématique visé de la caractéristique bivalente de ces logiciels.

Toutefois, nous en évoquerons dès maintenant, sous forme d'hypothèse, un corollaire relatif à la communication :

Face à un logiciel regroupant les deux types de communication, l'utilisateur investit plus spécialement l'un d'entre eux. Dans la plupart des cas, il s'agit du type II (selon sa modalité, c'est-à-dire le sous-type représenté dans le logiciel).

Notons d'ailleurs, sans en discuter ici la pertinence, que ces logiciels font souvent intervenir un score ou une limitation du temps susceptible de recentrer l'utilisateur sur l'objet du problème.

Ainsi D. Butten déclare qu'un score a été mis en place dans le logiciel TACHE pour limiter l'emploi de l'instruction «E».

A PROPOS DES TACHES RELATIVES AU CONTENU DE L'ACTIVITE

Nous examinerons les tâches relatives au contenu de l'activité essentiellement dans trois directions :

- participation de l'élève à la définition de l'activité.
- caractérisation de la tâche globale relative au contenu : statut de ce qu'effectue l'élève avec un logiciel.
- validation de l'activité (à ce propos, nous nous interrogerons sur le rôle de l'erreur dans une activité faisant intervenir un logiciel).

Ces divers points seront aussi l'occasion de cerner les limites de la créativité mathématique attachée selon certains à l'usage du logiciel LOGO.

I- PARTICIPATION DE L'ENFANT A LA DEFINITION DE L'ACTIVITE

La première partie de la recherche a établi que cette possibilité était offerte exclusivement par les logiciels à communication II, ceci à des degrés divers : autrui peut assumer la définition complète de l'activité et son expression (logiciels du type C, communication II- ou II+) ; le logiciel peut aussi ne lui permettre qu'un complément de définition portant sur l'objet de l'activité (logiciels de type B du sous-type 3.2, communication II-I), ou sur la démarche de résolution (logiciels de type B du sous-type 3.1, communication II-I).

Toutefois, nous n'excluons pas qu'avec des logiciels à communication de type I, l'élève se dérobe à l'activité imposée en en concevant implicitement une autre. Nous dirons, en tant qu'hypothèse, que cette éventualité est hautement probable avec les logiciels des sous-types I I-II* et I I-II* du type A, en raison de la communication de type II qu'ils proposent pour la réalisation d'un objet intermédiaire :

Les logiciels à communication I du sous-type I-II, engagent l'élève à redéfinir l'activité qu'ils imposent, en une activité portant sur un objet intermédiaire, ceci en raison d'une communication de type II spécifiquement disponible pour la réalisation d'un tel objet : il y a déplacement de l'objet de l'activité.

Des expérimentations confirment cette hypothèse que nous avons faites a priori :

- Nous proposerons, lors de l'examen de la tâche relative à la validation de la réponse, un indice susceptible de justifier qu'avec le logiciel RECT, l'activité dominante était celle d'affichage de la grille à l'écran qui précédait l'activité numérique imposée par le logiciel.

- Citons D. Butlen et C. Lethielleux à propos de leur logiciel nommé DECOU, ayant pour objectif le calcul de produits s'appuyant sur des découpages de grilles rectangulaires à l'écran :
"Pour l'ensemble des élèves, nous constatons que le but de l'activité est perçu comme le découpage et non comme un moyen de connaître le nombre total de croix."
("Cahier de didactique des mathématiques numéro 25₂" page 16).

Suite à ce préalable, nous étudierons les modalités de la tâche de définition de l'activité par l'élève pour les logiciels dont le contexte le permet directement, c'est-à-dire pour les logiciels à communication II.

1a Le principe du "projet" : origine et positions diverses

ORIGINE

Nous avons évoqué en page 18 le «projet LOGO» associé au logiciel LOGO, qui revendique pour l'enfant la possibilité d'un apprentissage autonome, c'est-à-dire sans instruction (au sens de "sans consigne") : dans cette perspective, l'enfant définit son propre projet, dont la réalisation est censée l'amener à découvrir en partie le «micromonde» mis à sa disposition. On trouve là, pour le domaine de l'emploi de moyens informatiques dans l'enseignement, l'origine de ce que nous appelons le principe du «projet». Nous n'oublierons pas qu'aujourd'hui, celui-ci a tendance à se réactualiser dans les derniers développements de l'Intelligence Artificielle (langages déclaratifs, «micromondes» autour du logiciel PROLOG).

⇔ POSITIONS DIVERSES

Pour certains, le principe du «projet» est une référence : si le logiciel l'autorise, l'enfant doit travailler par «projet», le choix de l'enseignant se portant d'ailleurs plus volontiers sur un tel logiciel. La participation de l'élève à la définition de l'activité se veut alors complète et doit se traduire par la conception d'un projet.

La définition du terme «projet» n'est jamais explicite : le mot apparaît dans les textes.

Chez des auteurs, dont S. Papert, le «projet» se caractérise par le fait que l'enfant (ou un groupe d'enfants) se le donne. Toutefois, cette

pseudo-définition se nuance très vite : ainsi, dans son ouvrage "L'ordinateur à l'école", G. Bossuet indique que l'adulte, — nommé «facilitateur» et non «enseignant» —, peut intervenir, sans préciser toutefois de quelle manière :

"Le facilitateur doit donc intervenir dans le choix des projets pour qu'ils soient, à un instant donné de l'expérimentation, à leur portée." (page 88)

D'autre part, si des écrits présentent tout d'abord le «projet» comme l'unique méthode de travail valide, les mêmes écrits apportent aussitôt des correctifs à leur déclaration d'intention : G. Bossuet parle de «projets» et de «contrats» ("L'Ordinateur à l'école", page 84), C. Berdonneau d'«exercices», de «problèmes» et de «projets», puis de «projets intégrés» et de «projets libres» ("Recueil de pratiques pédagogiques autour de LOGO", page 38).

A l'inverse, pour d'autres le principe du «projet» est contestable : ils se démarquent du «projet LOGO» en préconisant la définition de situations didactiques par l'enseignant.

Dans ce cas, la participation de l'élève à la définition de l'activité ne diffère pas de ce qu'elle peut être au cours de séquences sans ordinateur : elle dépend des décisions de l'enseignant.

A. Rouchier insiste dans son article "LOGO et les contenus d'enseignement et de formation" sur la nécessité d'une mise en œuvre didactique appropriée pour que les enfants connaissent une véritable expérience de mathématisation ; D. Guin mène à l'IREM de Strasbourg, une recherche faisant intervenir LOGO, où le point de vue retenu est celui du choix des situations par l'enseignant.

Nous adoptons cette position : les recherches en didactique, cette recherche et notamment ce chapitre constituent une argumentation en sa faveur. En particulier, ce qui suit prétend cerner le principe du «projet» en tant qu'illusion idéologique, les oppositions déjà signalées EAO VS LOGO ou enfant répétiteur, enseigné VS enfant créateur trouvant des expressions équivalentes en activité donnée par l'enseignant VS projet de l'enfant, ou encore non motivant VS motivant.

1b Définition de l'activité et principe du «projet»

Nous examinerons successivement la possibilité effective d'une participation de l'enfant à la définition d'une activité sur la base du principe du «projet» avec LOGO, puis avec les autres logiciels à communication de type II.

Nous ne nous poserons pas ici la question de la pertinence d'une telle méthode eu égard à des acquisitions d'ordre mathématique.

⇔ EXISTENCE ET ILLUSION DU «PROJET» AVEC LOGO

Il est incontestable que le logiciel LOGO offre la possibilité de définir une activité. Cela signifie-t-il pour autant l'existence de projets d'enfants? Nous avons observé directement ou consulté des documents relatant des «projets» LOGO d'enfants ou d'adultes. Leur caractéristique commune est l'identité des sujets : maison ou carré se retrouvent fréquemment, si ce n'est systématiquement. Comment expliquer qu'une liberté se traduise en un tel manque de diversité? Nous répondrons à cette question par d'autres questions accompagnées d'hypothèses.

• Quelle est la réalité de l'offre de définition de «projet» en contexte scolaire, fut-il expérimental?

L'élève est-il vraiment l'auteur du «projet»? Nous n'oublions pas qu'un adulte est présent dans la classe. Appeler celui-ci «facilitateur», comme le fait G. Bossuet nous paraît relever de l'idéologie, celle-là même de "l'enfant créateur" qui préside au principe du «projet».

Citons G. Bossuet dans "L'ordinateur à l'école", pages 76 et 77 :
 "Nous choisirons pour désigner la personne qui assiste les enfants dans leur utilisation de l'outil informatique le terme utilisé par C. Rogers de *facilitateur*.
 Le *facilitateur* est celui qui peut, grâce à ses connaissances et à son expérience, *aider* l'élève dans sa démarche, et *faciliter* son *autoformation*."

L'adulte est formateur par ses actes auprès des enfants, ne serait-ce que par sa présence, ou encore par le choix des primitives ou procédures prédéfinies qu'il met à leur disposition, ou encore

Signalons que dans un article de la revue "Education et Informatique" (n° 22 et 23) J.L. Zimmermann, C. Nidegger, A. Giordan du Laboratoire de didactique et épistémologie des Sciences (Faculté de psychologie et des Sciences de l'Education Université de Genève) remettent en cause eux aussi la réalité du statut de «facilitateur» en explicitant l'obstacle que constitue la formation même de l'adulte.

Par conséquent, il peut exister un contrat implicite sur le sujet de l'activité. La notion même de contrat didactique l'assure. Nous aboutissons ainsi à une hypothèse, raison possible à l'identité des thèmes :

Le «projet» de l'enfant est (au moins pour partie ?) celui de l'enseignant, que ce dernier soit intervenu explicitement ou non pour sa définition.

• L'élève est-il en mesure de répondre à l'offre de définition de projet qui lui est faite?

Tout d'abord, avoir un projet nous semble devoir être lié à un besoin : on ne définit pas gratuitement un projet. En contexte scolaire, quelle nécessité l'enfant satisfait-il lorsqu'il se conforme au principe du «projet» ?

La seule réponse vraisemblable nous semble être un désir d'affrontement à la machine : l'élève est vainqueur s'il réussit à faire exécuter son «projet». Outre le fait que nous sommes alors loin d'un objectif mathématique, une telle motivation survivra-t-elle aux nombreuses implantations de micro-ordinateurs (en particulier depuis le plan Informatique Pour Tous) qui banalisent le phénomène ?

Nous résumerons en une hypothèse qui traduit une autre explication possible à l'identité des thèmes de «projet» :

L'élève ne recherche pas réellement un sujet d'activité, ne fait pas réellement de «projet» \ tenant compte implicitement de l'influence de son environnement, il se donne pour but ce qui se présente d'emblée à son esprit.

Son «projet» peut par là-même constituer une façon inconsciente de tester sa supériorité sur la machine en consistant en la réalisation de dessins qu'il considère comme élémentaires, à savoir une maison, un carré,

A l'appui de cette hypothèse, remarquons que l'élève abandonne souvent son «projet» initial lorsqu'il commet une erreur qui en perturbe l'accomplissement : ainsi, une maison peut devenir une usine ou tout autre chose.

L'abandon du projet par modification (dans le meilleur des cas) peut être lu comme un signe de sa vacance.

D'autre part, définir un projet n'est pas une tâche évidente : l'enfant possède-t-il les connaissances pour la résoudre ?

La résolution du «projet» fait intervenir directement des connaissances (communication avec le logiciel, contenu mathématique) que l'on sait être limitées chez l'enfant de l'école élémentaire.

Indépendamment de cela, la mise au point d'un projet requiert une maîtrise d'ordre méthodologique qui n'est pas innée.

Ces deux facteurs nous conduisent à une hypothèse en deux temps :

- Les connaissances de l'élève réduisent le champ des projets possibles.

- L'élève adapte la machine à ses capacités intellectuelles du moment : il définit un «projet» immédiatement appréhendable, dont la maîtrise lui paraît acquise compte-tenu de ses savoirs.

Ceci constitue une raison supplémentaire à l'identité des thèmes abordés : une maison, un carré, c'est facile à définir, et a priori, à réaliser.

• L'offre de définition de «projet» est-elle toujours accueillie positivement par l'élève?

On constate chez des enfants (et chez des adultes) une lassitude face à l'activité LOGO, comme dans d'autres domaines.

La difficulté de la tâche, dès que l'on dépasse les premiers graphismes, peut l'expliquer.

En conséquence, nous proposerons une hypothèse relative à la synonymie entre activité donnée par l'enseignant VS projet de l'enfant, et non motivant VS motivant.

La motivation attachée au principe du «projet» n'est pas prouvée. L'élève peut ne pas avoir envie d'exploiter la proposition de définition de l'activité, ne serait-ce qu'en raison de la complexité des tâches qu'il aura à résoudre.

⇨ LE PROJET ET LES LOGICIELS A COMMUNICATION II AUTRES QUE LOGO

D'après nos travaux, le logiciel LOGO n'est pas le seul à permettre a priori la définition de «projets».

Cette possibilité se retrouve intrinsèquement avec l'ensemble des logiciels à communication II.

En effet, elle repose sur l'existence des systèmes d'instructions de production. Ceux-ci autorisent une production par sa description (dans le cas qui nous intéresse, c'est-à-dire celui des logiciels à sémantique procédurale) : ainsi, tout texte cohérent pour sa syntaxe correspond à un objet qui est susceptible d'être un projet. Signalons au demeurant, que cette liberté a priori consentie peut être oblitérée par le fait que le logiciel propose lui-même une activité (surtout en sous-type 3.1 II-1).

Certains observateurs contestent le point de vue précédent. Ils avancent qu'avec LOGO l'enfant conçoit des projets, tandis qu'avec d'autres logiciels que nous classons en communication II (comme PICTOR, communication II-), il n'en est rien. Nous récusons cette affirmation qui, selon nous, n'établit pas la réalité des projets d'élèves, mais mesure plutôt la difficulté relative de la tâche suivant le logiciel à communication II en cause. Ce qui peut engager à prendre pour un «projet» ce qui n'en est pas un, c'est la complexité de la tâche : les systèmes d'instructions proposés par le logiciel sont à cet égard déterminants.

Considérons une activité de dessin d'une maison à l'écran. Avec LOGO, la tâche peut s'avérer complexe, l'élève devant fournir une description de la maison en tenant compte du logiciel utilisé (tâche de communication et tâche d'ordre géométrique) : la réalisation prend du temps, est le cas échéant l'occasion d'essais multiples.

Par contre, avec PICTOR en mode dessin, il lui suffit de dessiner sur l'écran avec le crayon optique comme il le ferait sur du papier, ou encore de décomposer le graphisme en formes géométriques qui s'afficheront par la donnée de paramètres : dans le premier cas surtout, la tâche est élémentaire et essentiellement de nature matérielle (usage du crayon optique) ; la réalisation est quasi immédiate.

Toutefois, avec LOGO comme avec PICTOR, le «projet» de maison peut n'être qu'une illusion.

Nous ne reviendrons pas sur les réserves faites à propos du «projet» avec LOGO : elles s'appliquent globalement aux autres logiciels à communication II, même si des nuances que nous allons évoquer peuvent être introduites.

⇨ LE «PROJET» COMME PARTICIPATION EFFECTIVE DE L'ÉLÈVE À LA DÉFINITION DE L'ACTIVITÉ

Rappelons que nous ne discutons pas ici la pertinence du principe du «projet» eu égard à l'activité mathématique ; nous nous contentons d'essayer de préciser des conditions permettant à l'élève d'avoir une réelle participation à la définition de l'activité sous forme de «projet».

Lorsque le logiciel détermine partiellement l'activité et permet d'apporter un complément de définition portant sur son objet (sous-type 3.2 II-I), il n'est pas impossible qu'à défaut d'avoir un «projet», l'enfant puisse réellement participer à la définition de l'activité : la tâche est moins importante, il ne s'agit pas pour lui de peupler un univers vide, mais d'apporter des précisions à propos d'un domaine déjà cerné par le logiciel. Toutefois, nous nous interrogeons quant à l'influence du degré de complexité de la communication proposée par le logiciel eu égard à une participation effective à la définition de l'activité.

L'observation d'enfants de CM travaillant avec le logiciel MOSAÏQUE (sous-type 3.2 II-I) nous paraît le confirmer : les élèves n'affichaient pas n'importe quelle mosaïque, mais décidaient de ce qu'ils souhaitaient obtenir et persévéraient dans ce but. Toutefois, la communication dans ce logiciel, bien que de sous-type II-I, est particulièrement encadrée jusqu'à s'exprimer en questions attendant des réponses "Donne le code de ...".

Le «projet» nous apparaît donc plus plausible avec certains logiciels à communication II-I.

Avec l'ensemble des logiciels à communication II, le vécu de l'élève en matière d'usage d'un micro-ordinateur peut s'avérer décisif en engendrant des comportements diversement appropriés à la conception d'un «projet». En effet, l'enfant qui a déjà employé LOGO en contexte scolaire est susceptible d'avoir perçu un élément du contrat didactique que nous formulons en une hypothèse :

Avec un micro-ordinateur et LOGO, l'activité demandée par l'enseignant se nomme «projet». Utiliser un micro-ordinateur, faire un «projet» n'est pas un jeu : il s'agit d'un travail qui n'est pas aisé, qui suit certaines règles.

Pour étayer cette hypothèse, nous mentionnerons les difficultés que l'enfant a pu rencontrer au cours de la réalisation des pseudo projets que sont les «exercices», «problèmes», «contrats» (C. Berdonneau, G. Bossuet) que l'enseignant a pu lui demander de traiter.

Dans le cas où la forme de travail retenue par le maître consistait en séquences didactiques, l'hypothèse reste globalement valable.

En conséquence, si par la suite, on présente un logiciel à l'enfant, il peut reconduire les termes du contrat.

C'est ainsi que nous avons observé que des élèves face à PICTOR (communication II-) après des activités LOGO se donnaient une activité relativement précise, contrairement à d'autres enfants sans vécu informatique antérieur qui se contentaient de crayonner l'écran.

De la même façon, il n'est pas impossible que la perception du contrat par l'élève permette dans une certaine mesure la conception d'une activité avec LOGO par l'élève, sur la base du principe du «projet».

⇨ CONCLUSION

Les logiciels à communication II donnent la possibilité de définition d'activités.

Il nous paraît illusoire que l'élève puisse réellement gérer cette offre.

Toutefois, des conditions particulières imputables au logiciel (définition partielle de l'activité, communication facilitante) ou au vécu scolaire informatique de l'élève peuvent lui permettre d'y mieux répondre.

Une question fondamentale, que nous traiterons maintenant, est posée : la définition de l'activité par l'élève (ou encore, le «projet» effectif) est-elle une méthode de travail valide?

1c La définition de l'activité par l'élève est-elle une méthode de travail valide?

⇒ PERTINENCE DE LA METHODE DANS LA CLASSE, DANS LA SCOLARITE

Nous nous interrogerons d'abord sur la viabilité d'une telle méthode de travail avec une classe de vingt-cinq à trente élèves.

On ne se trouve plus dans les conditions particulières offertes par les contextes expérimentaux (la plupart du temps, peu d'enfants et plusieurs adultes), lieux de développement du principe du «projet».

N'oublions pas qu'il s'agit de favoriser l'acquisition de savoirs : avec une classe ordinaire, comment l'adulte, — qu'il s'intitule «facilitateur» ou enseignant —, gèrera-t-il la diversité des activités ? En particulier, on connaît la difficulté qu'a l'enfant à s'intéresser à ce sur quoi il n'a pas été actif : dès lors, en supposant même qu'un contenu de formation (mathématique ou de communication) puisse apparaître commun à plusieurs «projets», comment s'établiront les phases de communication, de preuve, d'institutionnalisation ? C'est pourquoi, nous pensons que laisser à chaque élève (ou presque) la faculté de définir l'activité, implique l'hypothèse suivante :

Adopter le principe de la définition de l'activité par l'élève se fait au détriment des phases de communication, de preuve, d'institutionnalisation.

La solution du «projet de la classe» («projet intégré» selon C. Berdonneau) qui infirmerait cette hypothèse paraît peu recevable, à moins d'une mise en œuvre répondant à une règle de cohérence en prévision des moments collectifs : un corollaire ne serait-il pas alors l'aliénation de la liberté de définition de l'activité ... ?

Nous terminerons en situant la problématique du principe du «projet» dans la scolarité : peut-on envisager que l'élève soit à même de travailler par projet de la maternelle à l'université, lorsque l'on sait par ailleurs l'identité des thèmes qui sont retenus ? Ne peut-on craindre l'enlisement, la désaffection ? Et enfin, quel est le gain pour l'apprentissage mathématique ?

⇒ DEFINITION DE L'ACTIVITE PAR L'ELEVE ET CONTENU DE FORMATION

Le recours exclusif à la définition de l'activité par l'élève comme méthode de travail suppose la croyance en un contenu spontané, consubstantiel au logiciel employé. Nous généralisons ici à l'ensemble des logiciels à communication II ce que dénonce A. Rouchier au sujet de LOGO :

"Il y a eu, et il y a encore, un certain nombre de partisans de LOGO pour lesquels il semble que la géométrie, les mathématiques sont consubstantielles au système; autrement dit, un enfant placé devant un système LOGO découvre, doit découvrir, avec un minimum d'interventions extérieures, une théorie mathématique à travers les problèmes qu'elle résout et qui l'éclairent."
("LOGO et les contenus d'enseignement et de formation", A. Rouchier)

Nous partageons l'analyse de A. Rouchier qui signale qu'une telle croyance s'appuie — d'une part sur une simplification abusive de la conception de l'apprentissage, — d'autre part sur une conception assez spontanéiste de la mathématisation et des conditions de sa pratique en situation scolaire.

Comment imaginer en effet que l'élève puisse être en mesure de définir la "bonne activité", celle qui se révèlera adaptée à sa connaissance, mais aussi susceptible de lui faire acquérir de nouveaux savoirs en mathématique ou en matière de communication ? Les recherches en didactique des mathématiques, le chapitre relatif aux tâches de communication, comme la suite de notre propos ne nous semblent pas permettre de l'envisager.

1d Conclusion : qui doit assurer la définition de l'activité avec un logiciel à communication II?

L'argumentation développée nous engage à répondre que l'enseignant doit définir les situations qu'il proposera à la classe.

Nous montrerons au cours des chapitres à venir que les logiciels à communication II permettent d'engendrer un grand nombre de problèmes au cours de situations didactiques : l'enseignant peut préciser ces dernières en fonction d'objectifs d'apprentissage mathématique ou de communication.

Les avocats du principe du «projet» prétendent qu'un «projet», c'est quelque chose à faire, pas à apprendre. Nous dirons qu'il en est de même avec un problème, au vrai sens de ce terme.

L'élève, la classe peuvent participer, si ce n'est à la définition stricte du problème, du moins à la compréhension de son énoncé ou encore à son ajustement, — comme pour d'autres activités sans ordinateur.

Préserver une bonne utilisation didactique supposera que l'on laisse l'élève résoudre le problème avec les moyens mis à sa disposition, notamment avec les moyens logiciels spécifiques, c'est-à-dire, les instructions de production.

Les phases de l'apprentissage menacées par le principe du «projet» (communication, preuve, institutionnalisation) interviendront en leur temps.

Au demeurant, le comportement d'enfants que nous avons observés contredit l'équivalence entre projet VS problème et motivant VS non

motivant :

- Des élèves en activité de «projet» LOGO ne se trompent pas d'objectif : toute leur attention est dirigée sur l'exactitude de la programmation, et non sur le graphisme, objet du projet. En ce sens, on peut dire que la motivation réside dans le problème de programmation qui se pose.
- D'autres élèves travaillant à résoudre en LOGO un problème posé par le maître ne semblait pas moins motivés que les précédents.

En fait, les tenants du «projet» nous paraissent plus défendre le terme «projet», que le «projet» lui-même en tant qu'activité originale : appeler une activité «projet» dispense de la dénommer «problème» ; ainsi, on évite la référence au système d'enseignement et on préserve la thèse, — faut-il dire le mythe —, de l'enfant autonome et créateur. Une question se pose alors : qui tire bénéfice du recours au vocabulaire «projet», «facilitateur» ? L'enfant, ou l'adulte ?

Nous terminerons en évoquant l'article de G. Brousseau, "Apports de l'informatique à l'enseignement des mathématiques, à la didactique, et à la formation des maîtres".

Sans vouloir discuter ici le contenu de ce texte, ce qui se situerait à la limite des objectifs de notre recherche, nous notons que les positions prises vont contre le «projet», jusqu'à préciser l'intérêt, selon l'auteur, d'une dévolution de l'activité par le logiciel :

"L'arrivée des micro-ordinateurs dans les classes ouvre dans ce domaine de grandes possibilités d'amélioration : la partie de ces situations complexes qui consiste à recevoir de nombreuses décisions et à leur appliquer des feed backs diversifiés peut le plus souvent être confiée à l'appareil, de sorte que leur présentation aux élèves va s'en trouver considérablement simplifiée. Le maître peut la découvrir comme le fera l'élève sans qu'il soit nécessaire de la décrire, de l'expliquer et de justifier pour qu'il puisse la concevoir. Il peut à volonté reprendre le point de vue de l'enseignement et examiner les variables et les variantes à la lumière de son expérience ou des résultats des recherches. Le savoir visé s'en trouve beaucoup mieux défini et les rôles des élèves et du maître dans son acquisition mieux connus."
("Apports de l'informatique à l'enseignement des mathématiques, à la didactique, et à la formation des maîtres", G. Brousseau, 21 Juin 1984).

Nous conclurons en nous prononçant contre la définition de l'activité par l'élève. Une telle pratique nous semble utopique et peu propice à favoriser des acquisitions : nous ne la concevons que rarement, mais pourquoi pas, à l'occasion de séquences dont l'objectif spécifique serait la problématique de définition d'activités ...

2- STATUT DE L'OBJET PRODUIT AVEC UN LOGICIEL

Nous examinerons le statut de l'objet produit au cours d'une activité avec un logiciel, d'abord succinctement dans le cas d'un logiciel à communication I, puis de façon plus détaillée dans celui d'un logiciel à communication II.

2a Statut de l'objet produit avec un logiciel à communication I

Nous nous référons à la description que nous avons faite de la communication de type I : avec un logiciel à communication I, l'élève transmet des réponses sur la base de ses savoirs, pour satisfaire aux questions posées.

Par conséquent, l'objet produit au cours de l'activité est à strictement parler un ensemble de réponses ; par extension, il n'est pas impossible que l'enfant se livre à des inductions sur la base du dialogue qu'il a entretenu avec le logiciel.

Pour cerner le statut de l'objet que constituent l'ensemble de réponses et les éventuelles inductions, nous nous placerons dans le cas où le contenu scientifique du logiciel est irréprochable. L'hypothèse suivante peut alors être retenue :

Avec un logiciel à communication I, la tâche est d'ordre mathématique. L'objet susceptible d'être produit au cours de l'activité (ensemble de réponses et inductions), qu'il satisfasse ou non aux questions posées, est également d'ordre mathématique.

Argumentons cette hypothèse :

Avec un logiciel à communication I, l'élève ne définit pas l'activité ; il répond aux questions posées par le concepteur du logiciel. Or, nous nous intéressons à des logiciels à contenu mathématique : par suite, la tâche est d'ordre mathématique, que l'enfant sache ou non la résoudre.

Un logiciel à communication I, n'offrant généralement pas à l'élève de moyens de redéfinir cette tâche, l'activité sera d'ordre mathématique. Par conséquent, si l'élève répond, l'objet produit, qu'il constitue ou non une réponse exacte au problème de référence, a toutes les chances de l'être aussi.

Tout ceci suppose bien sûr que l'énoncé du problème soit sans équivoque.

En outre, une réserve doit être faite pour les logiciels à communication I, du sous type I-II. Nous avons vu qu'il n'était pas impossible que l'activité se déplace sur la réalisation d'un objet intermédiaire en mode commande. Dans ce cas, le contexte spécifique au mode commande que nous analyserons au paragraphe 2b peut exister ; en particulier, la tâche peut cesser

d'être d'ordre mathématique.

De plus, dans le contexte le plus courant à la date où nous écrivons, le logiciel ne se contente pas d'infirmer les réponses, mais fournit si besoin est une réponse exacte : un contrôle d'ordre mathématique de l'objet produit intervient.

En outre, avec un logiciel à communication I, au mieux, l'élève appréhende les informations que le concepteur du logiciel y a mis (questions, aides diverses, messages de validation, ...) : celles-ci sont d'ordre mathématique. Or, les inductions pratiquées par l'élève ont lieu à partir de ses propres représentations du problème, mais aussi à partir et par confrontation aux instructions émises par le logiciel. C'est pourquoi, dans les conditions décrites, avec un logiciel qui corrige les réponses et au contenu mathématique irréprochable, on peut avancer qu'elles reposent en partie sur des faits mathématiquement corrects.

En résumé, nous dirons que, avec un logiciel à communication I, sans prétendre qu'il y ait nécessairement construction de savoir, la tâche et le produit de l'activité de l'élève (réponses, inductions), - s'il n'est pas inexistant -, sont d'ordre mathématique.

2b Statut de l'objet produit avec un logiciel à communication II

Avec un logiciel à communication II, la situation diffère de celle explicitée à propos de la communication I, aussi bien pour la caractérisation de l'objet produit, que pour le statut de la tâche et de l'objet.

L'objet produit par l'élève au cours de l'activité est à strictement parler ce dont il a commandé l'affichage au moyen des instructions de production que le logiciel met à sa disposition ; par extension, l'enfant peut se livrer à des inductions sur la base de ce qu'il a obtenu de la machine.

A propos du statut de la tâche, rappelons qu'un logiciel à communication de type II autorise l'élève à prendre part à la définition de l'activité.

Par conséquent, deux contextes peuvent exister : l'activité est définie par l'élève (à l'exception peut-être en sous-type 3.1 II-1, où l'intervention de l'élève concerne la démarche), ou par l'enseignant.

Dans le premier cas, rien n'assure que la tâche que se fixe l'enfant soit d'ordre mathématique : si l'on ne retient pas l'idée de mathématiques consubstancielles au logiciel employé, on peut envisager l'existence de logiciels et d'activités n'impliquant pas de tâche mathématique réelle.

Dans le deuxième cas, nous supposons que la situation retenue par l'enseignant implique une tâche d'ordre mathématique, ceci de son point de

vue. C'est pourquoi, dans ce contexte comme dans celui où l'élève se fixe lui-même une tâche mathématique, rien n'est assuré du point de vue de ce dernier : l'exposé concernant le statut de l'objet produit (ou encore, de l'activité), veut spécifier ce qu'il en est en fait pour l'élève, dans tous les cas de figure.

A propos du statut de l'objet produit avec un logiciel à communication II pouvant impliquer des activités géométriques, nous ferons l'hypothèse suivante :

Indépendamment du mode de définition de l'activité, indépendamment du statut de la tâche :

- l'objet produit peut ne pas avoir de statut mathématique : il peut n'être qu'un dessin, qu'un graphisme.

- l'objet produit peut conduire l'élève à un savoir mathématique incorrect, car induit de faits mathématiquement inexacts.

Nous restreignons notre hypothèse aux logiciels pouvant impliquer des activités géométriques, car à la date où nous écrivons, ils sont exemplaires de la communication de type II. Il conviendrait d'envisager une généralisation.

Nous insistons sur le fait que les deux alinéas de l'hypothèse nous paraissent fondés, que la définition de l'activité soit le fait de l'élève ou de l'enseignant, et que la tâche soit ou non d'ordre mathématique. Néanmoins, suivant le contexte, des nuances peuvent exister, nuances que l'argumentation que nous allons développer, précise.

Tout d'abord, nous dirons que fondamentalement, ce qui s'affiche à l'écran est un graphisme.

Toutefois, pour obtenir l'affichage en question, l'élève doit effectuer une description, à l'aide des instructions de production : il y a donc activité d'explicitation d'un objet (nous y reviendrons lors de l'examen des spécificités des logiciels) menant à l'affichage d'un graphisme ; celle-ci peut mettre en jeu des éléments mathématiques.

En outre, suivant son aspect, le graphisme peut avoir une connotation mathématique (un carré, un cercle, un assemblage de polygones).

Enfin, la désignation qu'en fait l'élève, souligne le cas échéant cette référence au domaine mathématique (un carré plutôt qu'un carreau, un assemblage de polygones plutôt qu'une maison, un cercle plutôt qu'un rond ...). Au demeurant, si c'est le maître qui signifie l'activité, il peut le faire à l'aide d'un vocabulaire courant (rond, carreau) ou mathématique (cercle, carré).

C'est pourquoi, quelle que soit la primitivité des systèmes d'instructions présents dans le logiciel, en fonction de l'énoncé de l'activité, de l'allure du dessin, du contexte en général, l'élève peut percevoir dans le contrat didactique la possibilité, voire la nécessité d'affecter à la production graphique un statut mathématique.

Donnons quelques indications complémentaires sur l'existence de cet élément du contrat didactique :

- Il n'est pas nécessaire que le contrat didactique comporte l'élément d'association d'un statut mathématique à la production :

Avec les logiciels FRISE ou MOSAIQUE, l'enfant peut se contenter de réaliser un beau dessin, en organisant les positions relatives d'éléments de base : sa production ne concerne pas les isométries du plan.

- Répondre à une tâche d'association d'un statut mathématique au graphisme affiché ne nous paraît pas lié à la "nature graphique ou géométrique de l'exercice ou du projet".

Nous avons repris entre guillemets la terminologie employée par C. Berdonneau :

"Nous désignons par exercices ou projets graphiques ceux qui mettent en jeu la capacité du périphérique (écran, tortue ou coccinelle) d'effectuer des dessins, lorsque l'attention portée aux propriétés géométriques nécessaires à la réalisation du tracé reste secondaire. Dans le cas contraire, nous qualifions ces projets de géométriques.

Il n'y a bien sûr pas de frontière très nette entre ces deux types de projets ; c'est souvent l'enchaînement entre différents projets successifs qui déterminera s'ils sont graphiques ou géométriques, bien plus que l'objet lui-même du projet."

("Recueil de pratiques pédagogiques autour de LOGO", C. Berdonneau, page 55).

Il est troublant de constater la classification de projets que fait C. Berdonneau, après qu'elle ait indiqué à juste titre que l'objet du projet n'est pas déterminant pour le statut de l'activité : les projets "géométriques" sont généralement descriptibles par un terme géométrique, à l'inverse des projets "graphiques".

Nous dirons que, si le vocabulaire "graphique" et "géométrique" souligne bien le problème posé, nous ne pensons pas qu'il le résolve : préciser le statut de l'activité nous semble dépendre du contrat qui s'instaure et de la capacité, - que nous examinerons bientôt -, de l'élève à y satisfaire.

- La façon dont est formulée l'activité (par le maître ou par l'élève) peut constituer un facteur influant sur le contrat.

L'élève est plus directement conduit à associer un statut

mathématique (exact ou inexact) à sa production par la consigne «Fais construire un carré sur l'écran», que par la consigne «Fais réaliser ce dessin sur l'écran», le dessin demandé étant celui d'un carré.

Toutefois, la notion même de contrat didactique interdit de sceller définitivement le statut de l'activité sur le recours à un vocabulaire spécifique à un cadre donné ; en particulier, la désignation de l'objet en termes géométriques par l'enseignant n'est en rien une caution.

Nous poursuivrons en disant que si le contrat didactique exige de l'élève l'affectation d'un statut mathématique à l'objet produit, il n'est pas évident que celui-ci soit en mesure de le remplir de façon satisfaisante du point de vue mathématique :

- L'élève peut attribuer un statut mathématique inexact au graphisme dont il a obtenu l'affichage :

- Pour reproduire avec LOGO un dessin représentant un carré, des enfants de CE2 ont utilisé des rotations d'angle 89° (TD 89). Nous trouvons là une preuve de la production d'un graphisme, et non de l'objet mathématique désigné par le terme «carré». S'il n'est pas question de carré, on peut estimer qu'il y a réussite, compte-tenu de la finesse du quadrillage de l'écran. Dans le cas contraire, nous sommes en présence d'une tentative d'associer un statut mathématique faux à un simple graphisme.

- La situation ci-dessus s'est répétée avec des instituteurs en formation, à qui l'on avait demandé de réaliser avec LOGO des polygones réguliers (avec un objectif d'étude de la mesure de leurs angles).

Associant un statut mathématique à chaque graphisme obtenu, un élève instituteur a établi le tableau suivant, incorrect pour le pentagone et l'heptagone :

Nombre de côtés du polygone régulier	Mesure de l'angle (en degrés)
4	90
5	70
6	60
7	50
8	45

- L'élève peut se livrer, en raison de l'attribution d'un statut mathématique inexact à un graphisme, à des inductions menant à des savoirs mathématiques incorrects :

- Appeler «carré» le graphisme aux angles de 89° cité ci-dessus,

autorise l'induction suivante : "tous les angles d'un carré mesurent 89° ..."

- Avec le logiciel FRISE, l'utilisateur peut employer la "machine n°1" (le groupe d'isométries conservant la frise est engendré par une translation) pour produire une frise admettant des symétries qui la conservent globalement ; dans ce cas, il risque de conclure que, seules des translations conservent globalement la frise affichée.

⇒ EN CONCLUSION

Nous espérons avoir montré qu'au-delà de la question du statut de l'objet, l'utilisation d'un logiciel à communication II pose le problème de la validation de l'activité et des acteurs de la construction du savoir visé.

Peut-il s'agir de l'affaire exclusive de l'élève, voire même du groupe d'élève, si l'on prétend à des acquisitions d'ordre mathématique ?

Sans attendre la partie de ce chapitre où nous traiterons de la validation de l'activité en général, nous dirons qu'associer à un graphisme un statut mathématique correct et construire sur cette base des savoirs exacts, dépend :

- du moment où l'élève arrête son activité et valide ou non sa production, par rapport au statut qu'il prétend lui donner.

- des connaissances mathématiques de l'élève.

Ainsi, un élève du cycle moyen qui "connait l'angle droit", validera sans qu'il puisse y avoir équivoque, en tant que carré, le graphisme correspondant au texte REPETE 4 (AV 20 TD 90) : il pourra retrouver ou induire des propriétés du carré sur des données exactes.

Par contre, avec un élève du cycle élémentaire sans connaissances relatives à l'angle droit, la situation est différente : rien, au niveau logiciel, ne lui donne l'assurance de l'exactitude du statut qu'il affecte au graphisme (et des inductions résultantes), qu'il produise celui-ci avec une écriture faisant intervenir 89° ou 90° .

En synthèse, nous situerons le statut d'une l'activité menée à des fins d'apprentissage géométrique avec un logiciel à communication II, indépendamment d'éventuelles interventions de l'enseignant.

Signalons que nous reprendrons plus en détail ce que nous allons exposer, lors de l'étude de la validation et des spécificités des logiciels.

On peut distinguer deux types d'activités :

• L'utilisateur doit afficher à l'écran une image ayant une relation avec des notions géométriques.

Il y a alors deux situations possibles :

- L'utilisateur connaît des propriétés géométriques de l'objet à produire :

Il les réinvestit dans la communication qui lui est proposée par le logiciel, en utilisant les instructions de production pour décrire l'affichage. De ce fait, il fait fonctionner ses savoirs mathématiques en interaction avec ses connaissances en matière de communication.

Son activité est essentiellement une activité de communication. Toutefois, faisant fonctionner ses savoirs mathématiques, il n'est pas impossible qu'ils les réordonnent, les précisent. Il n'est pas non plus improbable qu'il soit amené à faire des inductions : sans rien affirmer quant à leurs exactitudes, on peut remarquer qu'elles se fondent sur des faits corrects.

- L'utilisateur n'a pas de connaissances mathématiques préalables concernant l'objet à produire :

Il recherche un emploi de la communication mise à sa disposition par le logiciel, pour réaliser un graphisme à l'écran. Ce faisant, il exprime sa représentation de l'objet à produire (ajustée à la communication en cause) : il fait implicitement des hypothèses qui mettent en jeu des propriétés géométriques.

Son activité, définie par un objectif mathématique, reste essentiellement une activité de communication : rien ne permet d'affirmer qu'il puisse se construire un savoir mathématique correct : s'il lui est possible d'apprécier son graphisme, cela sera au niveau de la perception.

• L'utilisateur est confronté à un texte ayant un rapport avec des notions géométriques, rédigé dans une communication donnée, et cherche à en maîtriser le sens.

C'est la situation de connaissance du sens des instructions de production qui est ici généralisée.

Cette généralisation correspond à une problématique, celle des logiciels comme cadre langage. Elle sera examinée au cours du chapitre relatif aux spécificités des logiciels, mais nous indiquerons dès maintenant que l'utilisateur peut tenter la résolution du problème à deux niveaux :

- en confrontant, suite à la transmission du texte au logiciel, sa perception de l'image à l'écran à ses connaissances mathématiques, pour

induire une réponse ; il s'agit surtout d'une activité expérimentale, il n'y a pas de perspective de démonstration.

- en construisant une réponse directement à partir du texte : la question de la démonstration, incluant celle de la traduction des concepts dans les cadres concernés peut alors se poser. Il s'agit d'une activité d'ordre mathématique.

2c Conclusion

Avec un logiciel à communication I, l'élève est face à une machine qui tient les rôles ordinaires des différents membres de la classe (enseignant et élèves) : en général, le logiciel définit une tâche mathématique, la formule à l'écran et intervient dans l'activité, en particulier par des messages de validation.

Tout semble concourir à ce que l'activité de l'élève soit d'ordre mathématique.

La motivation de l'élève repose essentiellement sur son désir de vaincre la machine en gagnant au « jeu » qu'elle propose. Le vocabulaire employé dans ces logiciels est à cet égard assez significatif : les titres comportent souvent le terme « jeu » ou y font référence (INVASION DES NOMBRES, JEU DE BOOLE, GOLF, ...) ; les messages sont également révélateurs (utilisation des mots "gagné", "perdu", "score", "bonus", "malus" plutôt que "juste", "faux", "note", ...). Au demeurant, même lorsque le logiciel permet un jeu entre deux enfants, les observations montrent que les deux concurrents se liguent et s'entraident pour obtenir un maximum de points à chaque tour, et ainsi mieux vaincre la machine.

Les logiciels à communication I autorisent-ils, de par l'ensemble de leurs caractéristiques, la construction de savoirs ? Nous tenterons de fournir des éléments de réponse à cette question au chapitre des spécificités des logiciels.

Avec un logiciel à communication II, l'élève ou le maître peuvent intervenir dans la définition de la situation. La validation n'est pas logique. La tâche n'est pas nécessairement d'ordre mathématique, et même lorsqu'elle l'est a priori, rien n'assure que l'activité de l'élève le soit.

En l'absence d'intervention extérieure, de confrontation au savoir savant, de données culturelles suffisantes sur le sujet traité, nous dirons que, l'activité n'est pas d'ordre mathématique. En effet, Les mathématiques ne sont pas consubstantielles au système ; dans les conditions citées, l'action de l'élève ne débouche pas sur l'acquisition non équivoque de savoirs mathématiques.

Par conséquent, nous proposerons le corollaire suivant :

Les moyens didactiques usuels de l'apprentissage mathématique (en particulier, phases de communication relatives à l'activité, de preuve, d'institutionnalisation) restent à notre avis indispensables : la machine, le

logiciel ne les remplacent pas. Nous affirmons ici le rôle fondamental du groupe et de l'enseignant et caractérisons les logiciels à communication II comme outils didactiques, — dont l'intérêt sera évoqué au chapitre des spécificités.

La motivation de l'élève repose ici aussi sur un désir de vaincre la machine, non plus en gagnant contre elle, mais en lui faisant faire quelque chose dont autrui (lui ou le maître) a décidé. Ceci est repris par certains adultes au niveau de l'idéologie de l'enfant créateur. Nous leur opposerons l'argumentation que nous venons de développer ; elle nous paraît cerner une limite de l'œuvre de création attribuée à l'enfant avec un logiciel à communication II : il n'y a pas de créativité mathématique systématique, seul avec un micro-ordinateur et un logiciel, fut-il à communication II.

Nous nous inscrivons en faux contre cette déclaration de C. Berdonneau : "L'utilisation de la machine fournit aux élèves une information sur l'efficacité de leur raisonnement". ("Recueil de pratiques pédagogiques autour de LOGO". C. Berdonneau, en citation de "Une tortue dans une classe, une année d'expérimentation en Cours Moyen 2", Cahier PACIFIC n°1. 1983 C. Berdonneau, R.M Dumas.).

Toujours sur la base de notre argumentation, la thèse suivant laquelle un logiciel à communication II (LOGO ou un autre), permet un apprentissage autonome, la construction par l'enfant de son propre savoir, nous paraît peu recevable.

Nous ne récusons pas les logiciels à communication II, — ce sont des outils didactiques dont nous étudierons ultérieurement les avantages —, mais l'utilisation qui peut en être faite.

3- VALIDATION DE L'ACTIVITE

Nous avons vu en établissant la typologie que les logiciels traitent différemment la validation de l'activité. Dans cette diversité, le type de communication paraît déterminant : c'est pourquoi, nous examinerons la validation de l'activité, successivement pour les logiciels à communication I, puis pour ceux à communication II.

3a Validation de l'activité avec un logiciel à communication I

Les logiciels concernés sont ceux de type A.

Le descriptif que nous en avons fait dans la première partie de la recherche, précise que, à de rares exceptions près, un logiciel de type A émet un ou plusieurs messages relatifs à l'attendu conceptuel du problème posé.

Notre propos étant une mise en valeur des caractéristiques générales de la validation avec un logiciel à communication I, nous n'étudierons les différences entre logiciels de type A qu'en référence aux genres de messages.

Toutefois, signalons que le moment d'intervention du message peut varier : l'affichage peut avoir lieu immédiatement après une réponse de l'élève, ou au contraire être différé pour permettre plusieurs tentatives portant sur la même question, ou des essais sur des questions intermédiaires.

⇒ MESSAGES D'ÉVALUATION STRICTE

Nous dirons qu'un logiciel comporte un message d'évaluation stricte, s'il infirme, confirme et/ou affiche un corrigé de la réponse de l'élève.

D'après nos observations, la plupart des logiciels à communication I traite l'activité de l'enfant par un message d'évaluation stricte ; ceci nous conduit à une hypothèse, confirmée dans de nombreux cas, concernant le point de vue logiciel relatif à la validation :

La validation proposée par un logiciel à communication I (c'est-à-dire de type A) est, le plus souvent, réduite à une évaluation.

Par conséquent, nous dirons qu'il n'y a pas de validation de la part du logiciel : dans ces conditions, qui prend en charge la validation de l'activité ?

Nous employons le terme de « validation » au sens d'argumentation, de preuve.
Le problème de la validation d'activités mathématiques en ce sens a été particulièrement étudié par N. Balacheff.

Nous répondrons par une autre hypothèse :

Avec un logiciel à communication I, l'élève ne prend pas en charge la validation de son activité.

Nous argumenterons cette hypothèse, qui repose sur des observations, en exposant ce que nous pensons être des raisons à un tel comportement.

- La tâche de validation est délicate ; déjà, lors de situations n'utilisant pas de logiciels, l'élève ne la prend que difficilement en considération. Or, les logiciels à communication I autorisent, cautionnent et même risquent de renforcer cette tendance de l'élève : intrinsèquement, dans un logiciel à communication I, rien n'oblige, ni même, ne permet à l'enfant de mener une activité de validation.

En effet, la communication de type I a été caractérisée par l'existence de cycles «réponse» (cycles entrée utilisateur → instruction machine, comprenant la donnée de réponse par l'utilisateur et son évaluation par la machine). Par conséquent, nous compléterons tout d'abord une précédente remarque suivant laquelle un logiciel de type A concentre les rôles ordinaires des membres de la classe, en disant que son utilisation optimum est souvent individuelle.

Dans ce contexte, l'élève transmet à la machine, — et à personne d'autre —, une réponse ; puis, la machine affiche un message d'évaluation, que l'enfant constate ; le message est accompagné ou suivi d'une instruction visant à la poursuite de l'activité. Contrairement à un groupe d'enfant ou à un enseignant, le logiciel n'argumente pas, n'offre pas de moyens de le faire ; il n'y pas de contestation (autre qu'une éventuelle infirmation) qui inciterait à considérer une tâche de validation. L'absence de la phase de validation n'est pas reconnue comme un obstacle : l'activité prévue par le logiciel continue, au sens où les écrans suivants s'affichent.

Ainsi, l'élève ignore la tâche de validation, se contente d'une évaluation extérieure.

- Nous avons noté que lorsqu'il n'y a qu'une infirmation ou confirmation de la réponse (cas du logiciel PUCE), l'élève ne paraît pas chercher à savoir ce qui motive le message du logiciel, ni qu'elle est la réponse correcte.

- Plus révélateur encore, est l'absence de réaction de l'élève face à l'absence complète d'évaluation dans le logiciel RECT. Citons D. Butlen :

"Ils ne pensent pas, dans une large proportion, que la validation de leur réponse soit nécessaire. Si celle-ci devait être faite, elle le serait soit par le maître, soit par l'ordinateur." ("Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", page 96, thèse de 3^e cycle, D. Butlen).

Participant à l'expérimentation, il nous est possible de préciser les circonstances de la citation précédente : les questions posées lors de l'entretien ("Crois-tu que tu as répondu correctement? Qui peut le savoir?"), en raison de l'absence de réaction des élèves durant la passation, ont suscité une grande perplexité. N'obtenant pas de réponse, les expérimentateurs ont alors été conduits à faire des suggestions (le maître?, la machine?) qui ont été reprises par les élèves : pour eux, la question ne se posait pas.

- La communication de type I, par son fonctionnement d'ensemble, peut expliquer le comportement d'ignorance de la tâche de la validation par l'élève.

Un logiciel à communication I évalue : l'élève n'a a priori aucun intérêt à valider une activité déjà évaluée.

Un logiciel à communication I concentre tous les rôles ; en particulier, il définit l'activité et intervient dans sa gestion par des propositions directes. Par conséquent, l'élève agit en réponse aux demandes du logiciel ; tout concourt à

lui laisser penser que, s'il fallait valider, cela serait clairement indiqué : dans le «jeu» attaché à la communication I, l'élève n'est pas maître des règles, il se plie à celles qui sont édictées. Ainsi, même s'il n'est pas passif par rapport au «jeu», il peut l'être par rapport à l'activité.

• Avec un logiciel à communication I, il peut exister un déplacement de l'objet de la validation, tout particulièrement si une communication de type II est offerte pour la réalisation d'un objet intermédiaire.

Un logiciel à communication I établit une motivation de l'élève sur la base du jeu contre la machine : en ce sens, il peut autoriser un déplacement de l'objet de la validation. Si l'enjeu n'est pas le problème mathématique posé, mais la victoire contre la machine, il n'est pas impossible que, pour l'enfant, l'objet de la validation soit le respect des règles. Notons d'ailleurs que, dans le cas des logiciels à communication I, le contexte de «jeu», et non d'activité mathématique, nous paraît renforcer le peu d'intérêt qu'aurait l'élève à valider suite à l'évaluation conçue comme sanction du «jeu». Nous ne dénisons pas pour autant les perspectives offertes par certains jeux, avec ou sans ordinateur, pour certains apprentissages.

Pour les logiciels regroupant les deux types de communication, nous avons déjà émis en pages 155 et 156 des hypothèses suivant lesquelles l'utilisateur investit plus spécialement l'un des types de communication, généralement le type II, et est ainsi conduit à redéfinir l'activité. Nous les reprenons ici pour les étayer, comme pour en examiner les conséquences.

Lorsque le logiciel permet la production d'un objet intermédiaire en mode commande, l'élève peut le concevoir comme objet de l'activité, et corrélativement de la validation.

Le caractère gratifiant de la communication II (commander à l'ordinateur ce dont on a décidé, même dans une mesure limitée) peut l'expliquer.

Nous avons cité en page 157 des expérimentations faites avec les logiciels DECOU et RECT confirmant nos hypothèses ; nous ajouterons ici, que pour le logiciel RECT, l'absence de réaction face à l'absence d'évaluation nous paraît constituer un indice significatif de la dominance de l'activité d'affichage de grille à l'écran.

En admettant cette dominance, on comprend mieux le désintérêt de l'élève pour la validation de l'activité de dénombrement et sa perplexité face aux questions des expérimentateurs lors de l'entretien : de son point de vue, il a validé son activité puisqu'il a pu constater que l'affichage des grilles s'opérait correctement.

N'oublions pas non plus que le temps consacré à l'entrée de réponse (en mode réponse) est généralement très faible comparé à celui passé à la

production (en mode commande).

En résumé, avec RECT, mais aussi avec la plupart des logiciels permettant la construction d'un objet intermédiaire en mode commande (sous-types I I-II⁺ et I I-II⁻ du type A) :

- la phase "intermédiaire" est mobilisatrice : l'élève commande, construit avec les moyens logiciels (instructions de production) ; la durée de l'activité est importante.

- l'activité de donnée de réponse intervient en rupture : l'élève ne dispose d'aucun moyen logiciel pour résoudre la question, il doit effectuer une construction personnelle ; la durée de l'activité est moindre (qu'il y ait ou non une limitation du temps faite par le logiciel).

Nous ne nions pas pour autant l'utilité pour l'apprentissage, de la réalisation d'objets intermédiaires.

Nous voulons simplement souligner que, dans le cas informatique, il y a un risque de détournement de l'activité mathématique projetée : le faisceau de faits cités, imputable à la communication de type II, peut entraîner l'enfant à considérer le problème primitif sans rapport avec son action ; c'est donc cette dernière qu'il est susceptible de valider.

Remarquons également que, lorsque la production d'un objet intermédiaire en mode commande est optionnel, l'élève peut, pour les raisons déjà invoquées, y recourir même si cela ne lui est pas nécessaire : il y a possibilité d'activité gratuite, sans intérêt pour l'apprentissage, occasionnant peut-être même une déperdition.

Aucun élève n'a utilisé l'instruction R de RECT qui permet de donner une réponse avant l'affichage complet de la grille : nous nous demandons si la présence des grilles entières à l'écran était indispensable à tous les enfants?

D. Builen a incorporé dans TACHE un score dont l'une des fonctions est, selon ses dires, de limiter l'emploi de l'instruction E (affichage de points supplémentaires sur la grille masquée), c'est-à-dire en d'autres termes, la construction d'objets intermédiaires.

⇒ MESSAGES AUTRES QUE D'ÉVALUATION STRICTE

La teneur de ces messages peut consister, par exemple, en une autre expression de la réponse apportée par l'élève, en la donnée d'un score, en indications concernant la durée de l'activité ou de la réponse, ...

Ces différents éléments peuvent exister séparément ou regroupés ; ils accompagnent, le cas échéant un message d'évaluation stricte.

Les caractéristiques de la validation énoncées pour des logiciels de type A à évaluation stricte nous paraissent rester valables dans une large mesure ;

toutefois, des nuances peuvent être introduites.

Scores et compteurs de temps

Si le logiciel propose un score, il est possible que dans certaines circonstances (variables selon l'activité), le « jeu » contre le micro-ordinateur fonctionne positivement : l'enfant reconsidère son activité, engage une validation partielle, pour améliorer son score.

Les logiciels DIVIS et TACHE affichent un score.

Citons D. Butlen à propos de TACHE :

" les élèves, dans leur volonté de gagner contre l'ordinateur n'utilisaient la touche E, que dans un minimum de cas, préférant prendre (de façon consciente ou non) des risques en évaluant de visu plutôt que de perdre des points." "Notons que le jeu contre l'ordinateur a eu un aspect positif dans le cycle a)..."
 ("Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", page 131, thèse de 3^e cycle, D. Butlen).

Au contraire, dans d'autres conditions, le score peut avoir un impact négatif sur le problème à traiter et sa validation. En effet, on peut assister dans le contexte de « jeu » de la communication I, à une rédefinition implicite de l'activité : il s'agit, non plus de résoudre un problème avec un score optimum, mais exclusivement de faire le meilleur score.

Citons encore D. Butlen :

"Il semble que la volonté de perdre le moins de points possible ait eu un effet positif au cycle a) mais négatif au cycle b)." "Bien des élèves ont d'autre part manifesté un vif contentement lors de l'obtention à une grille d'un score supérieur à celui de l'ordinateur, toutefois comme nous le verrons au paragraphe suivant, l'élève ne cherche pas à comprendre les raisons de cette "victoire"."
 ("Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", pages 130 et 131, thèse de 3^e cycle, D. Butlen).

Le risque signalé ci-dessus est d'autant plus grand que l'activité proposée à l'élève constitue pour lui une véritable situation-problème. Pour justifier cette affirmation, nous examinerons le statut du score, et des implications à sa présence dans un logiciel.

Un score mesure un état de résolution du problème posé, en quelque sorte, la qualité de cette résolution.

Plus exactement, nous distinguons l'évaluation par une note de celle par un score :

– une note sanctionne l'exactitude de la réponse. Elle représente une évaluation finale, sommative ; il s'agit d'une évaluation normative.

– un score indique, en cas de réponse exacte au problème posé, la qualité de cette réponse ou de la procédure de sa construction. On peut dire que l'évaluation par un score tend vers une évaluation formative, ou s'apparente à une évaluation formative non explicitée.

Remarquons que l'évaluation par score était, avant l'arrivée des logiciels, peu introduit car vraisemblablement difficilement réalisable.

• Score et note nous paraissent présenter des avantages spécifiques, que nous croyons souhaitable de ne pas confondre : il est possible d'informer séparément sur l'exactitude et sur la qualité de la réponse.

Selon nous, tout score bien conçu devrait respecter son originalité en ne mélangeant pas ces deux aspects de l'évaluation.

C'est pourquoi nous reprocherons au score du logiciel TACHE de remplir les deux rôles : lorsque la réponse est fautive, le score est nul, alors qu'en cas de réponse correcte, il dépend du nombre d'appui de la touche E.

Par contre, le logiciel CLASSEMENT ALPHABETIQUE (Cedic/Nathan), dont l'intérêt principal nous semble être d'ordre mathématique (recherche de stratégie) différencie bien les deux aspects : un «malus» sanctionne les erreurs de classement (objet du problème), tandis qu'un «bonus» mesure la qualité de la stratégie employée.

• Toutefois, de l'intérêt même du score nous semble venir la difficulté didactique à son emploi.

Analysons le problème de ces évaluations formatives non explicitées, du point de vue de l'élève :

Le score optimum est un nombre que l'enfant doit atteindre en s'éloignant le moins possible d'un nombre témoin par valeurs inférieures ou supérieures. Le nombre témoin n'est pas nécessairement le score optimum ; selon le cas, il est connu ou non de l'enfant au début de l'activité. Le score de l'élève se construit soit par perte ou gain de points par rapport au nombre témoin.

- Avec TACHE, le nombre témoin est 10 ; le score optimum dépend de la grille traitée et s'obtient, comme le score de l'enfant, par soustraction du nombre d'emploi de l'instruction E.

- Avec MULTI ("Utilisation de l'ordinateur pour l'apprentissage d'un algorithme de calcul des produits", D. Butlen et C. Lethielleux, Cahier de didactique des mathématiques, n° 25-2), le nombre témoin est 0 ; le score optimum est 4 ; le score de l'élève s'obtient par addition au nombre témoin de un ou deux points, d'après une pondération des procédures employées.

- Avec DIVIS, le nombre témoin est 0 ; le score optimum dépend de la division euclidienne traitée ; le score de l'élève s'obtient par addition au nombre témoin du nombre de soustractions successives effectuées pour résoudre le problème.

En fait, il n'est pas improbable que, pour l'élève, la situation soit vécue très diversement, suivant le type de score : ne pas s'éloigner ou au contraire se rapprocher, par addition ou soustraction (le cas échéant, selon des poids différents), le plus souvent d'un nombre qui ne lui est pas précisé. On peut déjà relever une difficulté, celle de l'éloignement de ce système de calcul du système usuel de notation.

Nous ferons donc une remarque générale :

Des systèmes d'évaluation normative ou formative variés se mettent en place dans les logiciels à communication I, certains d'une assez grande complexité, certains mélangeant divers aspects de l'évaluation : il n'est pas évident qu'ils soient accessibles aux enfants ; peut-être constituent-ils un obstacle et surchargent-ils inutilement l'écran. Il serait intéressant d'étudier leur pertinence.

Sont concernés par cette remarque les compteurs de temps multiformes que l'on rencontre dans les logiciels à communication I (compteurs discrets, compteurs continus d'apparences diverses) : ils peuvent être assimilés à des scores, même s'ils ont d'autres fonction que nous examinerons au chapitre des spécificités.

Dépassons cette observation générale :

Réaliser le score optimum à partir du nombre témoin suppose une bonne, voire une complète, appréhension du problème traité : de ce fait même, comprendre pourquoi le score optimum n'est pas atteint nous paraît quasiment impossible pour l'élève qui ... ne sait pas l'atteindre ; il ne peut que constater et penser qu'il y a moyen de mieux faire. Certains contextes, parmi ceux où la situation proposée ne relève pas, du fait de ses caractéristiques, de l'égalité du nombre témoin et du score optimum, s'avèrent particulièrement gênants : comment l'élève peut-il maîtriser l'inégalité associée au problème posé, sans régler complètement celui-ci ? Dans ce cas, le score, perspective intéressante pour la remise en cause et la validation de l'activité, est au mieux ignoré, ou encore joue en négatif : l'élève peut se bloquer dans une recherche sans issue, ou résoudre de façon incorrecte le problème pour atteindre le nombre témoin.

Nous avons déjà vu que les élèves se satisfont avec TACHE d'un score supérieur à celui de l'ordinateur (alors que cela signifie une prestation moindre), et ne cherchent pas à comprendre les raisons de leur victoire.

D. Butten remarque aussi :

"Il est difficile de conclure sur l'efficacité de cet élément de correction, les élèves dans leur immense majorité (comme le prouve l'utilisation de la touche E) ne comprennent pas la signification réelle du score de l'ordinateur même s'ils répondent

oui quand on leur demande (et même après avoir reçu des explications de l'observateur)." ("Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", page 132, thèse de 3^e cycle, D. Butten).

Autre expression de la réponse de l'élève

Nous allons donner quelques indications tendant à montrer que, dans certaines circonstances, lorsque le logiciel propose une autre expression de la réponse de l'enfant, celle-ci peut constituer un élément mobilisateur d'une démarche de validation.

Les exemples dont nous avons connaissance concernent essentiellement un affichage de la réponse de l'élève dans un cadre différent de celui qu'il a utilisé ; néanmoins, il ne nous paraît pas impossible de conserver le cadre de travail de l'enfant.

Nous employons le terme «cadre» au sens de R. Douady.

Un concept peut intervenir dans plusieurs cadres. On connaît, par exemple, le cadre numérique, le cadre géométrique, le cadre graphique, ...

Le lecteur trouvera des informations relatives à la notion de «cadre», dans R. Douady, "Rapport enseignement apprentissage : dialectique outil-objet, jeux de cadres", (Cahier de didactique des mathématiques n°3 - publication IREM Paris VII), ou encore, R. Douady, "De la didactique des mathématiques à l'heure actuelle", (Mathématiques - didactique des) Encyclopédie Universalis, p. 885 - 889.

Cette autre expression de la réponse de l'élève peut intervenir de façon normative ou formative :

- Si la réponse de l'élève est exacte, il s'agit essentiellement d'une évaluation formative implicite, pouvant impliquer des démarches de reconsidération, de validation, en faisant intervenir un autre cadre ; de ce fait, cela est susceptible aussi de susciter des difficultés analogues à celles exposées pour les scores.

DIVIS est un exemple de logiciel donnant une autre expression de la réponse exacte de l'élève : à l'écran, s'affiche toujours le système définissant la division euclidienne, que l'élève a effectué par ailleurs selon une méthode de soustractions successives.

- Si la réponse de l'élève est inexacte, on peut penser qu'il s'agit d'une évaluation mixte (normative et formative), à condition que la réponse exacte ne soit pas fournie conjointement.

En effet, cette autre expression de la réponse de l'élève peut mettre en valeur la différence entre le problème posé et la réponse de l'élève, en

traduisant celle-ci dans un autre cadre ; la réponse finale n'étant pas donnée, la validation peut s'exercer. L'affichage de la bonne réponse ne présente pas cet avantage, d'autant que le « jeu » est alors terminé.

Le logiciel TACHE est un exemple : en cas d'écriture multiplicative inexacte, s'affiche en surimpression de la grille de points qui est l'objet de la question, une grille correspondant à la proposition de l'élève.

Signalons enfin, que suivant la situation, dans le cas de réponse inexacte de l'élève, une autre expression dans le cadre de départ pourrait se concevoir.

⇔ EN CONCLUSION

Les logiciels à communication I, surtout lorsqu'ils ne comportent que des messages d'évaluation stricte risquent d'accentuer, si ce n'est de provoquer, une tendance à l'oubli de la tâche de validation ou encore un déplacement de l'objet de la validation : il ne s'agirait pas alors d'un gain pour l'apprentissage.

Nous aimerions suggérer deux types de solutions au problème posé, que nous ne considérons pas comme exclusives l'une de l'autre :

- Construire des logiciels incitant fortement l'élève à travailler la validation de son activité. Les logiciels de type A affichant d'autres messages que ceux d'évaluation stricte constituent un début de réponse ; peut-être conviendrait-il de poursuivre des recherches en ce sens ?

- Construire des logiciels apparentables au type A n'émettant aucun élément d'évaluation du problème traité pour son contenu, mais permettant l'exercice de celle-ci en dehors de la machine. Certains peuvent considérer qu'il s'agit là d'un défaut (il n'y a pas de correction individuelle et immédiate de l'élève) ; nous trouvons pour notre part, eu égard à nos référents didactiques, que suivant l'objectif poursuivi, il peut s'agir d'une ouverture intéressante.

Nous mentionnerons comme base de réflexion des logiciels de type A tels que RECT et EVAMA, qui ne transmettent à l'élève aucune évaluation de l'activité pour son contenu.

Pour troublant que cela paraisse a priori, ceci est délibéré de la part des auteurs : nous allons tenter d'éclaircir nos motivations dans le cas de RECT (nous sommes co-auteur), et d'indiquer l'intérêt que présente EVAMA de ce point de vue :

- Nous ne sommes pas entièrement d'accord avec D. Butlen (co-auteur de RECT) qui déclare à propos de RECT : "Relevons pour terminer un défaut (non pas du matériel informatique) mais du logiciel testé : celui-ci ne permet pas de

valider (par une image), les réponses des élèves."

("Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", page 99, thèse de 3^e cycle, D. Butlen).

Nous dirons que l'image de la réponse aurait attiré l'attention de l'enfant sur le problème de la validation (premier type de solution proposé ci-dessus), mais n'aurait pas permis pour autant la découverte d'une méthode efficace de dénombrement, et de toute façon, pas celle de l'écriture multiplicative axb.

Il ne s'agit donc pas d'un réel "défaut", d'autant que le fait même que RECT n'affiche aucune évaluation relative au contenu (seul un message de dépassement du temps imparti pour la réponse peut apparaître) nous semble avoir autorisé une validation adaptée à l'institutionnalisation projetée de l'écriture axb, ceci à partir d'échanges avec l'ensemble de la classe. En effet, il était important pour l'institutionnalisation de cette notion que soit discutée l'efficacité des procédures de dénombrement, - ce qui paraît délicat au niveau logiciel. Ceci implique qu'ici, la validation intéressante ne devait pas porter sur la réponse mais sur la pertinence des stratégies, puisque dans la majorité des cas, les élèves ignoraient la désignation multiplicative et qu'il s'agissait donc de l'introduire.

Notons que si ce logiciel avait comporté une donnée de la réponse exacte sous forme canonique, cela aurait été inopérant pour le problème posé. Quant à l'affichage de l'écriture axb exacte, elle aurait conduit, dans ce cas d'usage du logiciel en amont de l'apprentissage, à un produit fonctionnant sur un contrat implicite du type imitation "j'apprends, j'applique".

- EVAMA (logiciel relatif à la proportionnalité, produit par le CATEN de Rennes), est présenté ainsi par D. Boisnard :

"Test court permettant de repérer à titre indicatif le niveau de l'utilisateur par rapport aux classes d'une taxonomie d'objectifs cognitifs. Logiciel permettant un cheminement linéaire avec prise en compte des résultats partiels et bilan final."

(Rapport d'activités 1984-85 du GRECO Didactique, Thème 3 Informatique, Sous-thème 2 Informatique et ingénierie didactique).

Dans ce logiciel, les activités demandées à l'élève dépendent des résultats qu'il a obtenus aux précédents exercices. Comme aucune question ne fait l'objet d'un affichage évaluatoire relatif au contenu et qu'un bilan final est transmis sous forme de types d'activités à entreprendre, nous pensons qu'il n'est pas impossible que cela induise chez l'élève un comportement différent, et en particulier, un questionnement relatif à la validation : nous espérons des résultats d'expérimentation sur ce point.

3b Validation de l'activité avec un logiciel à communication II

Les logiciels concernés sont ceux de types B et C.

D'après les caractéristiques générales des logiciels à communication II énoncées dans la première partie de la recherche, il n'y a pas d'émission de message relatif à l'attendu conceptuel du problème posé.

Ceci est cohérent avec le fait que ces logiciels offrent toujours au moins la possibilité d'apporter un complément de définition à l'activité ; le problème peut même être entièrement posé par soi-même ou autrui.

Ainsi, le logiciel n'évalue pas, ne valide pas l'activité mathématique.

Rappelons toutefois que certains logiciels du type B, spécialement parmi ceux du sous-type 3.1 II-I qui définissent l'activité complètement pour son objet, dérogent à cette règle en affichant une évaluation du problème posé, qui peut apparaître en première analyse superflue : avec DIVIS par exemple, l'utilisateur pourrait, comme avec les logiciels de type C, s'en passer et se sentir maître de la validation en visualisant l'écran.

⇒ POSITION DU PROBLEME

Nous n'oublierons pas qu'avec un logiciel à communication II, le statut de l'activité et de l'objet produit peut se révéler ambigu, voire ne pas être d'ordre mathématique : le problème de la validation de l'activité avec un logiciel à communication II se situe dans ce contexte. Autrement dit, de quelle validation s'agit-il ?

Pour répondre à cette question, nous nous référerons à la synthèse effectuée au sujet du statut d'une activité avec un logiciel à communication II (cf pages 173 -174), en l'étendant au cas de logiciels non géométriques.

Nous n'examinerons pas ici les problèmes de validation pour les deux types d'activités que nous y avons distingués : nous nous limiterons aux activités visant à l'affichage d'une production sur un périphérique de sortie ; ce n'est qu'au chapitre d'étude des spécificités des logiciels que nous aborderons le cas des situations de maîtrise du sens de textes.

Notre recherche a souligné que l'emploi d'un logiciel dans une situation d'enseignement fait coexister, quel que soit le type de référence du logiciel, deux activités : une activité de communication, et une activité d'ordre disciplinaire.

Dans le cas qui nous intéresse, nous avons dit qu'il y a un risque important de détournement de l'objet de l'activité : même si une production de type mathématique est définie comme but explicite, en l'absence d'intervention extérieure, l'activité fondamentale de l'élève peut rester de l'ordre de la communication. Toutefois, en particulier pour le domaine

géométrique, même si la production ne dépasse pas le niveau du graphisme, et si, par conséquent l'activité concerne surtout la communication, il est important de savoir comment l'élève se situe pour la validation de ce graphisme, ou mieux, si cela est possible, par rapport au contenu mathématique visé, tout particulièrement lorsqu'il n'a pas de connaissances spécifiques à son sujet.

Notre propos concernant essentiellement l'enseignement des mathématiques, nous évoquerons la validation, d'abord succinctement pour la composante communication de l'activité, puis plus longuement pour la composante ayant un rapport avec les mathématiques.

⇒ VALIDATION DE LA COMPOSANTE COMMUNICATION DE L'ACTIVITE

La validation de la communication est effectuée en partie par le logiciel : celui-ci évalue la correction du discours de l'élève, et émet le cas échéant un message syntaxique/sémantique.

Nous dirons que l'évaluation en question est minimale, au sens où, elle ne mesure pas la qualité du discours par rapport aux possibilités du système, mais seulement l'adéquation de celui-ci aux exigences de celui-là.

Nous renvoyons au chapitre relatif aux tâches de communication pour des interrogations relatives à l'influence sur les acquisitions d'ordre mathématique d'un discours utilisateur performant. Signalons seulement, qu'une bonne évaluation, du niveau d'écriture nous paraît nécessiter la confrontation aux autres (les élèves, le maître), et le cas échéant, la mise en œuvre de contrôles spécifiques (temps d'exécution, lisibilité du texte, possibilité de modification, d'adaptabilité, ...).

La pratique courante reconnaît une autre tâche de validation de la communication, celle de détection des erreurs logiques. Dans son acception la plus large, la question posée est celle de la preuve des programmes. Au niveau élémentaire où nous nous plaçons, cet aspect de la tâche rejoint le problème de validation de la composante «mathématique» de l'activité, que nous allons maintenant aborder.

⇒ VALIDATION DE LA COMPOSANTE «MATHEMATIQUE» DE L'ACTIVITE

• Généralités

Il s'agit de savoir si la suite d'instructions, voire dans certains cas, — par exemple, lors d'emplois spécifiques de logiciels de type C à communication II+ —, si l'algorithme conçu conduisent bien au résultat voulu.

Selon des conditions que nous allons tenter de préciser, la résolution de cette tâche aura statut d'évaluation ou encore de validation, ceci d'un graphisme ou encore d'un objet d'ordre mathématique.

Comme cela était le cas à propos du statut de l'objet produit avec un logiciel à communication II, ce que nous allons développer à propos de la validation nous semble indépendant du mode de définition de l'activité («problème» ou «projet»).

Nous ferons une hypothèse en trois parties.

Avec un logiciel à communication II :

– *L'élève se sent maître de l'évaluation ; il prend en charge une évaluation de son activité. Toutefois, rien ne permet d'affirmer qu'il engage une validation de l'activité ; peut-être même, n'est-il pas en mesure de le faire.*

– *L'évaluation repose sur la lecture (au sens large) sur le périphérique de sortie, de l'objet obtenu grâce aux instructions de production ; le plus souvent, il s'agira d'une visualisation à l'écran (périphérique prédominant aujourd'hui) de l'objet produit. Ainsi, dans le cas géométrique, il peut seulement s'agir d'une évaluation perceptive d'un graphisme.*

– *Les échanges entre l'élève et d'autres membres de la classe (élèves ou maître) à propos de la validation sont favorisés.*

Avant de commenter à proprement parler cette hypothèse, signalons qu'elle est largement admise, pour les logiciels de type C, et ceci en ces termes : l'élève valide une activité mathématique.

Pour notre part, nous la proposons différemment : nous l'appliquons aux logiciels de types B et C, et nous restreignons la faculté qu'aurait l'élève à valider son activité de façon autonome avec un logiciel à communication II, à savoir évaluation plutôt que validation, graphisme plutôt qu'objet géométrique, évaluation perceptive plutôt que raisonnée.

Nous distinguons, comme en communication I, validation d'évaluation : selon la première partie de notre hypothèse, avec un logiciel à communication de type II, la pratique la plus courante de l'élève est celle d'évaluation. Toutefois, nous verrons qu'il peut exister des tentatives de validation, en particulier, des pratiques locales de validation, qui ne sont pas pour autant la validation de l'activité dans son intégralité, c'est-à-dire par rapport à l'objectif

mathématique.

Enonçons tout d'abord plusieurs facteurs dont l'existence simultanée nous paraît contribuer à la reconnaissance par l'élève d'une tâche d'évaluation :

– Le logiciel n'évalue pas l'activité pour son contenu ; de ce fait, une question implicite est posée à l'élève : apprécier la distance entre ce qu'il voulait obtenir et ce qui s'affiche.

– Le rapport au micro-ordinateur en communication II (vaincre la machine en produisant ce dont autrui ou soi-même a décidé) empêche que la question soit ignorée.

– L'absence d'enchaînement systématique d'instructions par le logiciel renforce la probabilité que l'élève s'interroge sur la pertinence de son activité.

On peut remarquer que le contexte décrit est à l'opposé de celui évoqué pour la communication de type I, et semble plus prometteur.

Toutefois, la réponse qu'apporte l'élève à la question implicite de jugement de son activité, peut ne pas concerner une validation, mais seulement une évaluation qui, d'après la suite de notre hypothèse, peut n'être que de l'ordre de la visualisation : ainsi, dans le cas géométrique, il se peut que l'élève travaille au niveau du graphisme et compare au niveau de la simple perception.

Notons que, si notre discussion fait une place importante au géométrique, c'est en raison de la prédominance actuelle en communication II (surtout en type C) de logiciels portant sur ce domaine. Dès lors, en supposant que l'on sache mettre au point des logiciels à communication II, tout particulièrement de type C, proposant des activités numériques, qu'en sera-t-il de la validation ? L'évaluation graphique qui fonctionne comme une approximation d'une validation géométrique trouvera-t-elle un équivalent dans des situations numériques ? Est-ce souhaitable ? S'agira-t-il d'approximation numérique, d'intuition ? La question est ouverte.

En initialisation d'une éventuelle réponse, voyons dans quelle mesure, l'évaluation perceptive que l'on observe lors de situations géométriques peut déboucher sur une amorce de validation.

Si l'évaluation perceptive faite par l'élève de sa production est négative, nous dirons qu'intervient pour lui, un conflit que nous nommerons «machino-cognitif».

Nous le distinguerons du conflit socio-cognitif au chapitre d'étude des spécificités des logiciels ; nous mentionnerons ici, en tant hypothèse, ce que nous pensons être les principaux traits de son fonctionnement : l'élève constate

une disparité entre ce qu'il obtient et ce qu'il attend – ainsi, naît le conflit «machino-cognitif» – ; la nécessité pour lui, d'atteindre le but assigné, l'engage à remettre en cause sa production ; il est alors conduit à des pratiques de validation locale, exprimées par des îlots hypothético-déductifs qui rejoignent et impliquent des pratiques ou notions d'ordre mathématique ("j'avais fait cela ... alors ..." ; "Si je fais cela ... , alors ..."). Peuvent alors se constituer des théorèmes en acte.

Nous reprenons ici la formulation de G. Vergnaud.
G. Vergnaud appelle «théorème en acte», des "propriétés des relations saisies et utilisées par le sujet en situation de solution de problème", même si le sujet ne peut ni les expliciter, ni les justifier.
(G. Vergnaud - Revue Recherche en Didactique des Mathématiques Vol 2.2, 1981 ; L'enfant, la mathématique et la réalité, P. Lang Berne 1981).

En outre, selon la dernière partie de notre hypothèse, les échanges avec les divers membres de la classe ne sont pas exclus (recherche de renseignements, d'appréciation de la production) ; peut-être même, interviennent-ils nécessairement et participent-ils à ses validations ponctuelles.

Néanmoins, en dernier lieu, après le règlement de conflits successifs, nous dirons que la validation peut très bien ne rester qu'une évaluation perceptive d'un graphisme, dans le cas où l'élève n'a pas de connaissances sur le sujet mathématique abordé (des exemples comme le carré aux angles de 89° développés lors de l'étude du statut de l'objet produit paraissent bien l'indiquer). C'est pourquoi, dans ce cas, nous pensons qu'au mieux, on a affaire à une validation limitée de l'adéquation logique de la communication par l'élève : celui-ci apprécie la correction du texte qu'il a proposé pour produire le graphisme recherché.

Par conséquent, il importe de s'interroger maintenant sur les raisons de l'approximation de la validation de l'activité par l'évaluation non raisonnée, sur les conditions d'une validation d'ordre mathématique et sur les perspectives de recouvrement pour l'activité d'un statut mathématique : les instruments de la validation sont en cause.

- Un instrument de la validation : la méthode essai-erreur.

Avec un logiciel à communication II, on constate que l'élève procède souvent par essai-erreur. En effet, d'après le descriptif du type II de communication, l'utilisateur dispose à son gré des instructions pour commander.

C'est pourquoi, nous formulerons cette hypothèse :

La méthode essai-erreur est autorisée par les logiciels à communication II du fait de l'existence des instructions de production.

Nous situons tout d'abord la méthode essai-erreur.

Une méthode de travail par essai-erreur est revendiquée par certains comme méthode d'apprentissage permettant la construction de savoirs, mais aussi comme l'un des fondements de «l'apprentissage autonome».

Nous ne ferons pas cet amalgame idéologique : si l'engagement par l'élève de savoirs dans une problématique active d'essai-erreur nous paraît créer des conditions favorables aux acquisitions, ce n'est pas pour autant que l'apprentissage nous semble pouvoir être «autonome» ; l'ensemble du chapitre essaie de le prouver.

Remarquons d'ailleurs que des partisans d'un usage de LOGO conforme aux thèses de S. Papert, qui sont parmi ceux qui défendent l'«apprentissage autonome» sur la base du fonctionnement possible par essai-erreur, très curieusement, banissent le terme "erreur" de leur vocabulaire. Ainsi :

- Les messages syntaxiques/sémantiques du logiciel LOGO sont désignés dans les livres et manuels LOGO par «messages LOGO» ; il n'est pas question (permis?) de les appeler messages d'erreur, comme cela est le cas pour d'autres langages. Or, la présence parmi les «messages LOGO» d'un message signalant la définition d'une procédure ("Vous venez de définir ...", "... est défini") ne nous paraît pas suffisante pour expliquer cette rigueur terminologique.

- Lorsqu'un texte ne conduit pas au résultat souhaité, c'est-à-dire lorsqu'il y a une erreur logique de l'utilisateur (analyse du problème inexacte et/ou programmation incorrecte), le terme employé est «bug». Les arguments pour l'intraductibilité de ce mot en français laissent rêveur :

"Il est très difficile de traduire en français les termes de «bug» et de «debugging». Les vocables d'«erreur», de «faute» et de «correction» dénotent un sentiment de culpabilité qui n'existe précisément pas dans les termes américains et reflètent une attitude profondément différente à l'égard de l'apprentissage. Aussi bien seront-ils employés tels quels. D'ailleurs, le «bug» est davantage une *source d'erreur* qu'une erreur proprement dite." ("Jaillissement de l'esprit", S. Papert, page 35).

Nous voyons là une négation de l'erreur pour un soutien de l'idéologie de l'«enfant créateur» : celui-ci ne doit subir aucun affect, fut-ce au prix d'une perte de réalité.

Pour notre part, nous défendons le droit à l'erreur, jusque de son appellation, pour l'intérêt didactique qu'il présente (déséquilibre/rééquilibration des savoirs). Comme nous l'avons écrit plus haut, le recours à

des pratiques d'essai-erreur et par conséquent, l'éventualité de commettre des erreurs, nous paraît important dans la construction des savoirs. L'un des avantages des logiciels à communication II nous semble justement être cette possibilité qu'ils offrent de bâtir des situations où l'élève peut gérer ses erreurs, mais ceci, comme nous le verrons bientôt, dans la limite de ses connaissances et des apports du groupe.

Un fait nous paraît d'ailleurs démentir la transformation grâce à LOGO et au vocabulaire qui l'entoure, du vécu souvent négatif de l'erreur en vécu sans problème. Nous parlerons de l'emploi réservé de la pratique essai-erreur que nous avons observé aussi bien chez les adultes que chez les enfants : les utilisateurs de logiciels à communication II paraissent souhaiter que les essais et corrélativement les erreurs restent une affaire privée entre eux et la machine.

Ainsi, avec LOGO, l'utilisateur s'arrange fréquemment pour qu'un travail inachevé, au sens où il y a détecté une erreur et doit donc procéder à de nouveaux essais, ne soit pas visible par autrui ou conservé. Par exemple :

- Il met la tortue en mode "LEVECRAYON" pour lui faire exécuter un graphisme ; il voit l'évolution de la tortue, mais de cette façon, aucune trace ne figure à l'écran.
- Il donne très souvent un ordre d'effacement total de l'écran ("VE") presque aussitôt une erreur détectée.
- Il répugne à sauver un travail en cours sur disquette, même si l'enseignant le rassure quant à la place dont il dispose (et sur la fourniture d'autres disquettes, le cas échéant).
- Un cas extrême s'est trouvé en un adolescent qui éteignait et rallumait systématiquement le micro-ordinateur dès qu'il supposait l'approche du «facilitateur».

C'est pourquoi, nous émettons des réserves quant au vécu positif de l'erreur ou du «bug», que souligne par exemple G. Bossuet dans "L'ordinateur à l'école". En tout état de cause, le vocable «bug», pas plus selon nous qu'aucun changement d'ordre lexical, ne fera évoluer ce que l'apprenant s'autorise, et la didactique. Par contre, la possibilité offerte par un système informatique de respecter une condition de travail voulue par l'apprenant nous paraît prometteuse.

La pratique essai-erreur, telle que nous venons de la situer, est-elle un instrument de validation?

Nous ferons des hypothèses en réponse à cette question.

Avec un logiciel à communication II, la non-détection d'une erreur scelle aux yeux de l'enfant ce qu'il pense être la validation : l'élève évalue

son activité lorsqu'il cesse de fonctionner par essai-erreur.

Par conséquent, l'évaluation par l'élève, avec pour instrument la méthode essai-erreur, n'a de sens, que si ladite méthode peut être mobilisée.

Or, celle-ci ne peut l'être en l'absence d'une connaissance minimum de l'objet à produire : dans le cas géométrique, cette connaissance minimum peut être seulement de l'ordre de la perception (et il ne s'agira pas alors de validation mathématique) ; dans le cas numérique, le problème est ouvert (approximations sur la base de connaissances construites ?).

Rappelons que nous employons le terme «validation», au sens d'argumentation, de preuve ; l'évaluation ne comporte pas cette exigence.

Au sujet des problèmes didactiques concernant la preuve, il est possible de consulter N. Balacheff, "Preuve et démonstration", Recherches en didactique des mathématiques, vol 3.3, 1982.

Explicitons notre hypothèse :

Avec les instructions de production, l'élève peut exprimer sa représentation de l'objet.

L'évaluation faite par le logiciel de la description proposée par l'utilisateur consiste exclusivement en l'affichage correspondant au texte transmis.

Dans ces conditions, l'élève n'appréciera son travail que s'il connaît des éléments caractéristiques du problème traité. En l'absence de cela, — qui ne résoud d'ailleurs pas nécessairement le problème du point de vue de l'apprentissage mathématique —, nous dirons que l'activité est sans intérêt, peut conduire à l'élaboration de modèles erronés et être dommageable à la connaissance de la discipline.

Dans le cas de LOGO graphique utilisé à des fins géométriques, généralement, la connaissance perceptive des objets à produire ne fait pas défaut, surtout, lorsque ceux-ci sont présentés par des dessin et non par leurs dénominations : la méthode essai-erreur peut fonctionner ; par exemple, l'élève qui obtient comme affichage d'un triangle équilatéral, une ligne brisée comportant l'amorce d'un triangle isocèle non équilatéral n'évaluera pas positivement son activité.

Toutefois, toujours avec LOGO graphique, il n'est pas impossible que l'évaluation du graphisme même échoue : par exemple, dans le domaine de la simulation du fonctionnement d'objets techniques, une évaluation positive peut être faite alors que l'état de liaison du système n'est pas vérifié, parce que l'enfant ne dispose même pas de la connaissance perceptive de l'objet à l'étude.

Permettre l'expression de la représentation que l'on a du problème, et

l'évaluation au moins perceptive de celle-ci, dans une démarche par essai-erreur nous paraît constituer un intérêt d'ordre général des logiciels à communication II et de LOGO en particulier, intérêt sur lequel nous reviendrons.

Toutefois, nous avons vu que cela ne suffisait pas à assurer des acquisitions mathématiques.

Quelles conditions doit-on réunir pour un usage performant d'un logiciel à communication II ?

• Les connaissances intellectuelles, la culture mathématique comme instrument de validation

L'intérêt d'ordre général de LOGO, peut également être analysé en terme d'inconvénient, lorsque le corollaire de cet avantage est une croyance en la validation par l'enfant seul et l'apprentissage autonome.

En effet, s'il s'agit de mathématique, l'évaluation perceptive sur la base d'une connaissance perceptive ne peut suffire.

Nous reprendrons l'exemple du carré aux angles de 89° : déceler une erreur, valider au sens strict du terme, supposent une connaissance intellectuelle de l'objet. Dans le cas contraire, la validation sera illicite et/ou portera sur autre chose que l'objet primitif du problème (un dessin par exemple, dans le contexte géométrique).

Nos observations ne nous autorisent pas à partager l'avis de C. Berdonneau :
 "L'exécution faite par la machine permet à l'élève de se rendre compte de l'effet exact produit par sa description, et le cas échéant, il se trouve ainsi amené à la rectifier : cette modification se produit sans recours à une autorité magistrale -c'est l'enfant lui-même qui repère son bug, et cherche comment améliorer la réalisation, et non le maître qui indique une erreur-, et par suite sans connotation de culpabilité."
 ("Recueil de pratiques pédagogiques autour de LOGO", C. Berdonneau, page 164).

Nous dirons en tant qu'hypothèse :

– Avec un logiciel à communication II, il existe un seuil de connaissances mathématiques au dessous duquel la réflexion, les acquisitions ne se font pas ; ce qui semble flagrant pour le domaine numérique, est également vrai pour celui géométrique.

– La participation de l'enseignant, du groupe, à la phase de validation paraît indispensable dans la plupart des cas.

⇒ EN CONCLUSION

Les logiciels à communication II engagent à penser que la validation peut être l'affaire exclusive de l'élève.

Nous espérons avoir cerné le degré d'exactitude d'une telle position.

Nous aimerions présenter en synthèse quelques propositions concernant les logiciels, propositions éventuellement susceptibles de permettre une meilleure gestion de la tâche de validation par l'élève ; nous n'oublions pas pour autant le rôle fondamental que jouent le groupe et l'enseignant, dont nous parlerons séparément au cours de la conclusion, pour la commodité de l'exposé.

Il nous paraît souhaitable de proposer un logiciel à un élève, si celui-ci a les connaissances minimum pour l'aborder.

Il peut aussi être intéressant, et ceci sans supprimer au type II de communication son originalité (instructions de production) :

– de proposer un logiciel à communication II susceptible de vérifier l'exactitude de l'objet produit, par rapport au problème posé.

– de proposer un logiciel à communication II assorti d'éléments facilitant à l'élève un meilleur jugement de l'éventuelle approximation qu'il a réalisé.

Expliquons-nous :

• Certains logiciels en vérifiant l'exactitude de la production peuvent ne requérir qu'une connaissance minimum de départ.

Ce sont par exemple des logiciels mettant en jeu des systèmes d'instructions à indices faibles.

En effet, ils permettent des affichages par transmission de paramètres, par entrée de données dans des algorithmes prédéfinis, par appels de programmes prédéfinis : la production ne pourra être qu'intrinsèquement correcte du point de vue de la discipline.

Remarquons que l'exécution obtenue n'en est pour autant nécessairement la solution du problème, et que rien n'assure que l'enfant puisse valider.

Les logiciels n'autorisant qu'un complément de définition sur la démarche (sous-type 3.1 II-I du type B) appartiennent généralement à ce genre de logiciels. SIMUL (sous-type 3.1 II-I) semble intéressant par ses instructions qui ne réclame de l'enfant qu'une connaissance perceptive, tout en offrant un moyen d'exploration intellectuelle de l'objet graphique "grille rectangulaire de points". Si l'on en juge à l'absence de répétition du comptage du carreau du coin des grilles soumises au dénombrement, ou de façon plus générale aux résultats des

expérimentations que rapporte D. Butten dans "Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", ce logiciel intervient utilement pour préparer la présentation dans un jeu de cadres de la notation multiplicative.

Nous citerons aussi FRISE ou MOSAÏQUE (sous-type 3.2 II-I du type B), et de façon presque caricaturale PICTOR (sous-type 2 II- du type C) qui dispose d'une instruction «boîte rectangulaire» qui produit toujours un dessin de rectangle : on n'exige de l'élève qu'une association visuelle, le cas échéant la connaissance de la dénomination "rectangle", et la détermination du dessin par deux sommets opposés.

Mais cela permet-il de nouvelles acquisitions ?

• Pour les logiciels ne vérifiant pas l'exactitude de la production, en particulier pour les logiciels à systèmes d'instructions d'indices élevés, on peut concevoir une ou plusieurs instructions «Solution».

Tout spécialement dans le domaine graphique, cela permettrait peut-être de limiter les risques d'évaluation trop approximative.

L'appel de «Solution» afficherait le graphisme souhaité ou témoignant pour l'élève de ses réussites et erreurs.

Ceci suppose que les logiciels concernés, c'est-à-dire surtout ceux de type C, s'accompagnent d'une gamme d'activités prédéterminées.

Il y aurait donc définition au moins partielle de l'activité par le logiciel, ce qui autoriserait la génération des instructions «Solution». Les activités pourraient être paramétrables (adaptables) par l'enseignant ou les élèves. Notons que suivant le logiciel, l'adjonction d'énoncés et de fichiers «Solution», pourrait être réalisée par l'enseignant.

Mais, notre proposition est-elle plus pertinente qu'une consigne exprimée oralement, et surtout dispense-t-elle d'une validation avec la classe et de l'institutionnalisation des notions en cause ?

3c Conclusion

Le rôle de l'enseignant est fondamental.

Avec un logiciel à communication I, la validation peut être oubliée, ou déplacée.

Si l'enseignant ne doit pas être celui qui distribue magistralement le savoir, notre étude ne permet pas de penser que la tâche de le présenter incombera à l'outil.

Un logiciel à communication II ne nous paraît pas pouvoir régler en lui-même le problème de la validation : confier complètement la validation à l'élève face à la machine nous semble supposer une croyance idéologique en la

connaissance immédiate de l'enfant.

Nous ne retenons pas ce point de vue : les limites de la créativité mathématique de l'élève apparaissent clairement en matière de validation.

Agir en faisant des mathématiques expérimentales ne nous semble pas suffire aux apprentissages : la communication au sens ordinaire du terme, la validation, l'institutionnalisation des concepts restent indispensables. Les spécificités des logiciels à communication II pourront servir utilement l'apprentissage, si l'on reconnaît le rôle que l'enseignant a à jouer.

4- CONCLUSION : LA GESTION DES TACHES RELATIVES AU CONTENU DE L'ACTIVITE

Ce chapitre espère avoir mis en valeur plusieurs aspects intéressants des logiciels à communication II, mais aussi les problèmes qu'ils posent en égard à l'apprentissage mathématique.

S'ils apparaissent plus immédiatement proches de nos choix didactiques, ils ne constituent pas une solution en eux-mêmes.

La classe, le maître doivent intervenir : nous pensons que ceux-ci restent par leurs actes des éléments déterminants de l'acquisition de connaissances mathématiques.

Les logiciels à communication I ont certainement aussi un rôle à tenir.

Nous proposerons une étude des spécificités des logiciels selon une grille fonctionnelle didactique qui tentera de cerner la part que peuvent prendre les divers types de logiciels à l'enseignement.

Mais auparavant, nous allons examiner les conditions de transposition des concepts mathématiques avec des logiciels.

TRANSPPOSITION DE CONCEPTS MATHÉMATIQUES ET LOGICIELS

Si l'on admet l'hypothèse que nous avons faite, selon laquelle les deux seuls types de communication envisageables à moyen terme sont les types I et II, alors on admet aussi que la transposition de concepts mathématiques avec un logiciel s'exprimera dans ces types et seulement ceux-là.

Le cadre général étant ainsi défini, nous présenterons successivement dans ce chapitre nos remarques et hypothèses quant aux relations entre :

- types de concepts mathématiques et types de communication ;
- types de communication, types de logiciel et présentation des concepts mathématiques.

1- TYPES DE CONCEPTS MATHÉMATIQUES ET TYPES DE COMMUNICATION

La première partie de notre recherche montre qu'il existerait une corrélation entre :

- le type I de la typologie CONT et le type I de la typologie COM ;
- les types 2 et 3 de la typologie CONT et le type II de la typologie COM.

En outre, malgré une certaine évolution des logiciels depuis le début de notre travail, on constate que, encore aujourd'hui, ceux qui traitent de questions arithmétiques se trouvent surtout en type I de CONT, tandis que ceux qui s'intéressent à des concepts géométriques se classent plus fréquemment en type 2 de CONT.

C'est pourquoi, reprenant la question suivante posée lors de l'étude de la typologie CONT :

"Les faits suivants sont-ils contingents : type I et concept arithmétique, type 2 et concept géométrique".

nous la reformulerons ainsi :

Les faits suivants sont-ils contingents :

- Les logiciels à communication I traitent surtout de concepts arithmétiques.
- Les logiciels à communication II traitent surtout de concepts géométriques.

Les éléments de réponse que nous proposerons constituent une analyse des conséquences d'une situation que nous rappelons ici : le type II de communication est fondé sur l'existence d'instructions de production (ce sont elles qui autorisent le mode commande), tandis que le type I n'impose aucun préalable.

1a Type I de communication et concepts mathématiques

Il ne devrait pas y avoir corrélation entre type I de communication et concept arithmétique.

En effet, puisqu'aucun prérequis particulier n'est exigé, les logiciels à communication I (type A) peuvent a priori présenter des questions géométriques comme arithmétiques.

Les dernières parutions du commerce en font d'ailleurs la preuve : par exemple, citons le logiciel DROITE (Ed Cedic/Nathan), à communication I, qui interroge au sujet des droites du plan et de leurs positions relatives.

Les marques attachées au type I de communication (logiciels à mode réponse) seront présentes dans les logiciels de type A portant sur le domaine géométrique.

Ainsi, on peut faire l'hypothèse que la transposition de questions géométriques avec le type I de communication concernera essentiellement une approche terminologique de la géométrie, dont nous ne pensons pas qu'elle permette, en tout cas à elle seule, la construction de savoirs.

1b Type II de communication et concepts mathématiques

Pour les logiciels dont nous nous occupons (logiciels à sémantique procédurale), nous avons vu que les instructions de production correspondent à des objets ou opérations primitifs permettant la construction d'un objet complexe par sa description.

En outre, il va de soi que l'objet complexe en question est en rapport avec l'objet d'enseignement.

Par conséquent, le problème posé peut s'exprimer ainsi :

Quels objets ou quelles opérations mathématiques peut-on traduire en instructions?

ou encore :

Quelles questions mathématiques peuvent donner lieu à des définitions d'objets ou d'opérations qui soient primitifs d'autres objets et traductibles en instructions?

Nous limiterons notre tentative de réponse au cadre des mathématiques intéressant l'école élémentaire.

⇔ DOMAINE GEOMETRIQUE NON AXIOMATISE ET DOMAINE NUMERIQUE

Nous opposerons ces deux domaines.

L'école élémentaire est principalement concernée par une géométrie non axiomatisée.

Or, dans une telle géométrie, le travail s'effectue directement sur l'objet du problème, sans transformation préalable : le travail n'intervient pas sur des désignations de l'objet et il n'y a pas de recherche de modèle à effectuer.

En effet, on étudie par divers moyens des figures ; on en repère des invariants ; les opérations géométriques «transforment» des figures.

Ainsi, il s'agit essentiellement d'une géométrie physique expérimentale où les acquisitions sont surtout de l'ordre du savoir-faire (par exemple, savoir dessiner l'image d'une figure par une composition d'isométries, savoir produire des tracés spécifiques). Même si parfois, des raisonnements hypothético-déductifs sont localement nécessaires, sont surtout requises des pratiques de visualisation, construction, description.

A l'inverse, le domaine numérique présente d'emblée des désignations des objets d'étude (les nombres sont désignés dans un système de numération) et des opérations sur celles-ci.

Beaucoup de questions, dès l'école élémentaire impliquent la recherche de modèles adéquats avant tout traitement algorithmique (résolution de problèmes) ; les quatre opérations arithmétiques prennent leur sens en partie dans de telles activités.

On pourrait presque dire que l'élève doit étudier un système abstrait (si ce terme a un sens) très organisé.

⇨ PROSPECTIVE

Dans ces conditions, nous ferons une première hypothèse :

La corrélation actuellement constatée pour les logiciels à sémantique procédurale entre type I de communication et concepts arithmétiques, type II de communication et concepts géométriques, peut s'expliquer :

- Il est possible de définir "naturellement" des instructions de logiciels à sémantique procédurale, correspondant aux objets et opérations d'une géométrie physique expérimentale.

- Il est délicat de trouver parmi les éléments caractéristiques du domaine numérique des désignations et opérations primitives, susceptibles d'être des instructions.

Voici quelques exemples à l'appui de notre hypothèse :

- Les instructions des logiciels PICTOR, FRISE, MOSAIQUE, LOGO (type II de communication) correspondent à des objets d'exploration d'une géométrie physique.

- Les logiciels RECT et TACHE ne procurent pas un contre-exemple : en effet, leurs instructions concernent la production d'un objet graphique (grille rectangulaire de points), objet intermédiaire dans la résolution du problème numérique posé.

Toutefois, nous pensons que la corrélation existante peut être dépassée.

Nous ferons à ce sujet une deuxième hypothèse :

- Des savoir-faires numériques (assimilation d'algorithmes des techniques opératoires, mise au point de stratégies, raisonnement) impliquent des pratiques d'ordre algorithmique. Il ne paraît donc pas a priori impossible de réaliser des logiciels à sémantique procédurale et communication II (de type B ou C) qui considèrent ces questions.

- La construction par l'élève de connaissances numériques suppose de multiples interrogations concernant les relations qu'entretiennent les objets d'enseignement du domaine en question. Ne peut-on envisager des logiciels à communication II et sémantique déclarative qui s'intéresseraient à ce problème ?

Remarquons que rien ne semble devoir empêcher la construction

de logiciels à sémantique déclarative portant sur des questions géométriques.

Ainsi, une rééquilibration aurait lieu à plusieurs niveaux :

- Les concepts arithmétiques et géométriques seraient traités par chaque type de logiciels, avec son originalité propre.
- Savoirs et savoir-faires seraient pris en compte par chaque type de logiciels, et donc, pas uniquement par des logiciels à mode réponse (type A).

Nous terminerons par des exemples qui vont dans le sens de notre deuxième hypothèse :

- Le logiciel à communication II, DIVIS, vérifie la première partie de l'hypothèse : il met à disposition de l'utilisateur des instructions correspondant à des «opérations primitives» pour la mise au point d'un «objet complexe», à savoir une technique opératoire de la division euclidienne ; les instructions sont des soustractions permettant d'effectuer avec une efficacité variable suivant le choix qui est fait, la division proposée.

- Les logiciels de jeux de cible sont analysables en terme de logiciels à communication II et sémantique déclarative. Peu nombreux au début de notre recherche, ils se développent maintenant (travaux de membres de l'équipe de recherche "groupe IREM-CREEM"), notamment dans le domaine numérique. D'ores et déjà, on peut citer :

. Le logiciel EXPLO (J. Robinet et S. Hocquengheim) qui permet au niveau de la classe de 4^e, l'exploration du domaine des fractions par encadrements successifs.
 . L'ensemble de logiciels REPERAGES (P. Errecalde, S. Hocquengheim et J. Rogalski) qui s'adresse aux enfants de dix à treize ans pour l'acquisition des relations nombre-espace.

2- TYPES DE LOGICIEL, DE COMMUNICATION ET PRESENTATION DES CONCEPTS MATHÉMATIQUES

Nous allons préciser maintenant les liens évoqués au cours de la partie I de la recherche entre types de logiciels, types de communication et présentation des concepts.

Nous nous interrogerons également sur le caractère obligatoire de tels liens.

Nous examinerons successivement le cas des logiciels de type A, puis de type C, puis de type B.

2a Logiciels de type A (communication de type I)

D'après la première partie de la recherche, un logiciel de type A, c'est-à-dire à communication I, présente un seul concept à la fois, pour un de ses aspects, par une série de questions isomorphes impliquant généralement un seul type de procédure chez l'utilisateur.

Ainsi, il semblerait qu'un logiciel de type A mette en cause une unique question très délimitée par son objet et l'approche qui en est permise.

Autrement dit, d'après nos résultats :

Avec un logiciel de type A, la transposition didactique n'intervient pas pour un espace de concepts : elle ne place pas non plus le concept visé dans un jeu de cadres.

Rappelons que nous employons le terme «cadre» au sens de R. Douady.

Un concept peut intervenir dans plusieurs cadres. On connaît, par exemple, le cadre numérique, le cadre géométrique, le cadre graphique, ...

Le lecteur trouvera des informations relatives à la notion de «cadre», dans R. Douady, "Rapport enseignement apprentissage : dialectique outil-objet, jeux de cadres", (Cahier de didactique des mathématiques n°3 - publication IREM Paris VII), ou encore, R. Douady, "De la didactique des mathématiques à l'heure actuelle", (Mathématiques - didactique des) Encyclopédie Universalis, p. 885 - 889.

Nous ferons cette hypothèse :

Le type I de communication (mode réponse) induit fortement une telle expression de la transposition didactique ; toutefois, nous ne pensons pas qu'il l'impose.

La première partie de l'hypothèse nous paraît se comprendre aisément. C'est pourquoi, nous argumenterons surtout la dernière partie en citant quelques exemples de logiciels A non stricts qui constituent des ouvertures dans diverses directions :

- Les logiciels RECT et TACHE présentent des activités relatives à la désignation multiplicative dans un jeu de cadres numérique/graphique.

Remarquons que, la communication de type II intervient pour le cadre graphique (ce dernier sert à la réalisation d'un objet intermédiaire - une grille de points -, qui n'est pas directement le dénombrement, objet du problème posé).

- Le logiciel EVAMA que nous avons déjà mentionné questionne l'utilisateur sur la proportionnalité, mais ceci dans un champ conceptuel.

Plus précisément, en référence aux travaux de R. Gras, peuvent

servir de mots-clés pour sélectionner une situation outre trois classes d'âge, cinq indicateurs de niveaux opératoires, huit types d'activités, et diverses notions dont les descripteurs sont par exemple aire, volume, vitesse, échelle, pourcentage, ...
On trouvera plus de détails concernant ce logiciel dans "Rapport d'activités 1984-85 du GRECO Didactique, Thème 3 Informalique, Sous-thème 2 Informatique et ingénierie didactique".

- Le logiciel INVASION DES CHIFFRES (Ed Cedic/Nathan) pose plusieurs questions simultanément.
Il peut ainsi amener l'élève à organiser l'ordre de leur traitement pour une meilleure efficacité ; nous étudierons au prochain chapitre la particularité de ce logiciel du point de vue de la gestion du temps.

Remarquons qu'il a déjà été question de ces logiciels : ils sont de type A non-strict, mais ceci pas seulement pour les caractéristiques que nous examinons ici : l'éloignement du type de référence selon plusieurs critères est-il nécessaire ?
Ainsi, nous savons que RECT et EVAMA n'effectuent pas d'évaluation par rapport à l'attendu conceptuel, que RECT et TACHE ne présentent pas de consignes à l'écran.
Ces deux derniers logiciels construits à fin de recherche corroborent en outre l'idée que, si l'émission de consignes à l'écran est fortement cohérente avec le type I de communication (questions, mode réponse), elle n'est pas une obligation.

Nous compléterons en remarquant que la décision par le logiciel du contenu de l'activité, sans possibilité d'intervention autre qu'un éventuel paramétrage par l'enseignant avant l'utilisation en classe, est une caractéristique du type A strict, liée au type I de communication : quelles sont les questions dont les réponses doivent être évaluées ?

Par contre, nous venons de signaler que l'émission de consignes à l'écran, marque ordinaire du type A, n'est pas indispensable.

2b Logiciels de type C (communication de type II)

Selon le descriptif du type C, un logiciel de type C (communication de sous-types II+ et II- du type II) implique une infinité de problèmes non-isomorphes concernant plusieurs concepts.

Pour mémoire, nous donnerons la définition du terme «champ conceptuel» proposée par G. Vergnaud :

"Un champ conceptuel est un espace de problèmes ou de situations-problèmes dont le traitement implique des concepts et des procédures de plusieurs types en étroite connexion".

("Quelques concepts, quelques généralités, quelques références." Collectif ; Cahier de Didactique des mathématiques" n°5 ; Publication IREM Paris VII).

Autrement dit, d'après nos résultats :

Avec un logiciel de type C (communication de sous-types II+ et II- du type II), la transposition didactique intervient pour un, voire plusieurs champs conceptuels.

Nous soulignerons d'abord l'intérêt que nous y trouvons en citant J. Rogalski :

"Le point de fond est le fait qu'à la fois un concept est à l'œuvre dans une variété de situations et dans une situation donnée plusieurs concepts sont en jeu, ce qui conduit à rejeter l'étude de concepts isolés, identifiés à un type de situations."

("Quelques éléments de théorie piagétienne et didactique des mathématiques." J. Rogalski ; Cahier de Didactique des mathématiques" n°2).

Nous ferons maintenant une hypothèse explicative que nous étendons à l'ensemble des logiciels à communication II (type B et C) :

Le type II de communication peut exprimer la transposition didactique par champ conceptuel, ou tout au moins par plusieurs questions, et permettre des procédures diversifiées de résolution, en raison de l'existence des instructions de production.

La description générale que nous avons tentée de la communication de type II, au travers de divers systèmes d'instructions de production (cf p. 126 - 129), nous semble justifier cette hypothèse.

En effet, est mis en valeur le fait qu'avec les instructions de production, des possibilités combinatoires existent par choix de paramètres, d'objets, d'opérations.

Ce sont ces combinaisons potentielles qui permettent d'engager des questions diversifiées, voire de véritables champs conceptuels.

Il s'agit là, selon nous, d'une différence essentielle distinguant les logiciels de type B ou C des logiciels de type A.

On peut d'ailleurs recourir à la question théorique suivante pour tenter de caractériser rapidement la communication proposée par un logiciel :

"Est-il possible de traiter un autre problème que celui qui est posé (le cas échéant par le logiciel, en particulier en type B)?"

Donnons quelques exemples :

Grâce à leurs instructions de production,

- Le logiciel LOGO (type C) autorise la définition de nombreuses situations sur les formes géométriques, les transformations géométriques.

- Le logiciel DIVIS (type B) permet l'émergence de procédures variées relatives à la constitution d'une technique de division euclidienne par soustractions successives.

- Le logiciel FRISE (type B) met à disposition de l'utilisateur un espace de concepts relatif aux isométries du plan, aux groupes d'isométries admettant pour sous-groupe le groupe libre engendré par une translation.

Toutefois, il est évident que la capacité d'un logiciel à communication II à engager des questions multiples ou à autoriser des démarches variées de l'utilisateur dépend des caractéristiques des systèmes d'instructions qu'il comporte : nombre d'instructions, nombre de systèmes définissables, indices de ceux-ci comme témoins de la diversité des objets manipulables, ou de la primitivité des instructions.

Nous formulerons l'hypothèse suivante avec réserve :

Pour un objet donné, plus la description imposée par au moins un couple (objet, système d'instructions du logiciel) est détaillée, plus la capacité du logiciel en matière d'engagement de champ conceptuel est grande.

Cette hypothèse reste à prouver.

Néanmoins, elle peut s'envisager dans la mesure où la nécessité de fournir un texte très détaillé pour un objet peut aussi signifier une plus grande primitivité des instructions : en conséquence, celles-ci ne sont pas strictement liées à celui-là, et peuvent donc servir pour la production d'objets ou de procédures très différents.

Compte-tenu de ce que nous venons de développer, il est possible de préciser maintenant la signification de la propriété que nous avons attachée aux logiciels de type C de ne pas "décider du contenu de l'activité" : nous dirons qu'il s'agit d'une absence de prédétermination du champ parmi les champs possibles, ou des problèmes parmi l'espace de problèmes.

Ainsi, une analyse en terme de conséquence indirecte de la communication de type II est plausible.

Toutefois, y-a-t-il équivalence entre absence de décision sur le contenu par le logiciel et communication de type II?

Nous répondrons négativement par une hypothèse, qui contredit ce que le présent peut laisser croire :

Le défaut de décision par le logiciel du contenu à traiter est une condition suffisante, mais non nécessaire à la répertoriatio n d'un logiciel en communication II.

Précisons d'abord que nous ne parlons pas de l'émission des consignes.

Par conséquent, un logiciel comme RECT (communication de type I et absence d'émission de consignes) ne constitue pas un contre-exemple invalidant la condition suffisante : RECT décide du contenu à traiter, mais n'émet pas de consigne correspondante, empruntant simplement là une caractéristique extérieure de certains logiciels à communication II.

Par contre, les logiciels de type B montrent que la condition n'est pas nécessaire : ils décident du contenu à traiter, peuvent même l'afficher à l'écran, et sont à communication de type II.

Dès lors, est-il préférable qu'un logiciel décide ou non du contenu que l'utilisateur doit aborder?

En particulier, serait-il intéressant d'imaginer des logiciels de type C qui, sans perdre leur originalité, définiraient des situations-problèmes?

Nous avons déjà fait une proposition en ce sens à propos de la validation de l'activité. Conception et expérimentation de tels logiciels permettraient un meilleur jugement.

2c Logiciels de type B (communication de type II)

En référence au descriptif du type B, nous dirons que, tout en présentant une communication de type II, un logiciel de type B peut s'apparenter pour la présentation des concepts au type A ou au type C.

⇨ LE SOUS-TYPE 3.2 II-1

Les logiciels de type B du sous-type 3.2 II-1 sont proches du type C.

Ils impliquent généralement un espace de problèmes, voire un champ conceptuel.

Toutefois, les possibilités sont plus limitées qu'en type C. Notamment, l'habillage des notions ou du champ est réalisé, ce qui permet de dire qu'un logiciel de type B décide des problèmes à traiter.

Contrairement à LOGO ou PICTOR (type C) qui n'imposent quasiment aucune direction d'activités a priori :

- Dans le champ des isométries laissant globalement invariante une figure, le logiciel FRISE (type B) décide du problème, au moins

pour son thème : on ne peut construire que des frises.

- Le logiciel MOSAIQUE (type B) n'autorise que l'affichage de certaines mosaïques à l'écran.
Néanmoins, si avec un logiciel de sous-type 3.2 II-I du type B, le champ ou les activités au niveau de leurs thèmes est plus cerné qu'avec un logiciel de type C, nous n'oublierons pas que la communication de type II autorise des interventions extérieures : pour ce sous-type, des compléments de définition en particulier de l'objet du problème sont permises. Dans le cas des exemples cités, on pourrait vouloir réaliser telle mosaïque pour ..., utiliser tel système générateur de frises pour ..., ...

En conclusion, nous pensons qu'il n'y a pas de différence fondamentale pour la présentation des concepts mathématiques entre les logiciels de type C et les logiciels de type B du sous-type 3.2 II-I. Il s'agit plutôt de nuances quantitatives : espace des problèmes possibles plus ou moins étendu, apparente évidence de la définition du problème à traiter en raison d'un thème d'ensemble déjà déterminé.

⇒ LE SOUS-TYPE 3.1 II-I

Les logiciels de type B du sous-type 3.1 II-I sont plus proches du type A.

En consultant le descriptif de la typologie CONT, on s'aperçoit que la présentation des concepts est de même genre pour ces logiciels et pour ceux du type A : un logiciel du sous-type 3.1 II-I implique un nombre fini de problèmes isomorphes portant sur un concept ; le complément de définition est possible, mais seulement sur la démarche.

Sur ce dernier point, les logiciels du sous-type 3.1 II-I se distinguent de ceux du type A : grâce à la communication II, il est encore possible à l'utilisateur d'exprimer des procédures diverses pour la résolution du problème posé.

- Avec DIVIS, la recherche du quotient par soustractions successives peut se faire de différentes façons.

- Avec SIMUL, l'affichage d'une grille de points ne relève pas toujours de la même procédure (cf "Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", thèse de 3^e cycle, D. Butlen).

3- CONCLUSION

Les logiciels font intervenir les concepts mathématiques selon trois façons décrites par les trois types de la typologie CONT, dont deux surtout font références.

La communication est un élément déterminant :

- Les logiciels à communication I (type A) présentent un seul concept, généralement arithmétique, pour un de ses aspect, par une question.

Il y a fermeture sur le contenu, et vraisemblablement la démarche que peut avoir l'élève.

- Les logiciels à communication II (types B et C) impliquent des espaces de problèmes, voire des champs conceptuels, le plus souvent géométriques.

Il y a ouverture au moins sur la démarche (plusieurs procédures de résolution sont autorisées), et dans de nombreux cas sur le contenu.

Une évolution des logiciels n'est pas exclue.

Déjà, des logiciels d'un type empruntent des caractéristiques à un autre type.

Rappelons que nous avons tenté de cerner le degré de stabilité de notre travail en pages 117 - 120.

En outre, s'il est plus aisé de construire des systèmes d'instructions pour la transposition du domaine géométrique, nous nous souviendrons de l'hypothèse selon laquelle les deux types de communication peuvent exprimer la transposition des deux types de concepts mathématiques.

De plus, les deux types de communication, avec leurs originalités respectives, semblent pouvoir impliquer des champs conceptuels, comme des questions limitées.

L'ensemble de ces hypothèses tend à faire concevoir la communication proposée par un logiciel à l'utilisateur, comme variable didactique : en effet, il serait possible de la choisir dans certaines limites.

Encore faudrait-il apprécier en quoi le type de communication peut changer les rapports des élèves avec les notions en jeu dans une situation.

Une première approche des enjeux des deux types de communication a été faite au cours des chapitres précédents : nous avons étudié des aspects de l'influence de la communication, tant par les tâches nouvelles qu'elle suscite, que par son impact sur l'activité mathématique au niveau des tâches de contenu.

Il nous reste à nous interroger sur la spécificité de la présentation d'un champ conceptuel par logiciel, ou d'une question limitée par logiciel.

Pour quels objectifs recourrait-on à un logiciel déterminé ?

Le dernier chapitre, qui se veut une conclusion, examine ces questions.

SPECIFICITES DES LOGICIELS POUR L'ENSEIGNEMENT DES MATHÉMATIQUES

I- POSITION DU PROBLEME

A propos de l'informatique et de l'enseignement, deux questions sont généralement soulevées :

- L'informatique ou l'ordinateur présentent-ils un intérêt dans ou pour l'enseignement ?
- Tel logiciel fait-il usage des spécificités du micro-ordinateur ?

Exprimées de cette façon, ces questions sont-elles pertinentes ?

⇒ A PROPOS DE LA PREMIERE QUESTION

Formulée ainsi, un implicite peut y être attaché, celui d'une considération de l'informatique comme solution globale aux problèmes de l'apprentissage.

En outre, le sujet de l'interrogation, l'«informatique» ou l'«ordinateur», est trop large pour l'expérience que l'on a actuellement dans l'enseignement.

C'est pourquoi, nous formulerons la question différemment, en même temps que nous la compléterons :

Quel(s) changement(s) l'informatique peut-elle ou va-t-elle produire dans l'enseignement ?

L'emploi de l'ordinateur, ou plus exactement de logiciels peut-il bénéficier à l'enseignement ?

Y-a-t-il des différences entre :

- recourir à un logiciel :
- recourir à un manuel avec exercices (éventuellement corrigés) :
- organiser une séquence avec ou sans matériel.

Est-ce que l'on peut, avec un logiciel, organiser des activités :

- qui permettent un apprentissage (problème d'existence) ?
- qui permettent un meilleur apprentissage ?

L'ensemble de notre recherche, définissant des éléments d'une problématique de l'informatique-outil, espère engager une réponse.

Pour partielle que soit nécessairement celle-ci, ce dernier chapitre tente de compléter l'apport des précédents en proposant une étude fonctionnelle didactique, d'ordre général, des modes d'intervention possibles des logiciels (ce que nous désignons par le titre "spécificités des logiciels pour l'enseignement des mathématiques").

Nous ne nous préoccupons pas des coûts (en matériel, en logiciel, en formation des enseignants), ce qui, pour des décideurs institutionnels peut constituer un autre implicite de la question évoquée.

⇒ A PROPOS DE LA DEUXIEME QUESTION

Nous la remettons en cause : ce faisant, nous précisons que nous ne cherchons pas à trancher à propos de l'introduction de l'ordinateur, en particulier à l'école, mais à apprécier les potentialités de ce nouvel outil.

La deuxième question ne nous paraît pas pertinente.

Pour plusieurs raisons que nous allons exposer, nous dirons qu'elle ne détermine pas réellement une problématique de l'informatique-outil, aussi bien du point de vue de l'analyse des produits existants, que de celui d'une prospective de construction.

- Le matériel évolue rapidement.

Par conséquent, une évaluation de logiciels sur un critère de recours aux spécificités de la machine est vite dépassée.

En outre, l'imagination des développeurs de produits informatiques peut-elle s'exercer en référence à des matériels changeants ?

- La question posée n'est pas discriminante de logiciels.

Une formulation plus précise de la deuxième question serait :

Tel logiciel fait-il usage des spécificités du micro-ordinateur ?
Desquelles ?

Or, selon nous, tout logiciel exploite des spécificités du micro-ordinateur.

Par exemple, un logiciel d'enseignement programmé utilise, par l'intermédiaire d'un langage de programmation ou d'un langage-auteur, des caractéristiques de la machine : avec un logiciel d'enseignement programmé, il

est possible de proposer à l'apprenant des exercices répétitifs, ou encore une série d'items dont les réponses sont immédiatement et individuellement contrôlées ; un tel logiciel se sert de spécificités du micro-ordinateur, les activités qu'il autorise étant difficilement concevables de façon identique sans micro-ordinateur.

Le genre de logiciels cité en exemple peut satisfaire des enseignants proches d'une théorie behavioriste de l'apprentissage, tandis que d'autres, se retrouvant plus volontiers dans un modèle de type piagétien ne l'agrèeront pas.

En d'autres termes, chaque partisan d'une théorie de l'apprentissage peut justifier l'usage que fait un logiciel en adéquation avec la théorie dont il se réclame, des spécificités du micro-ordinateur.

Le critère n'est donc pas pertinent dans cette forme d'expression, car trop commun, trop général.

En résumé, baser l'analyse ou l'écriture de logiciels sur l'emploi de spécificités du micro-ordinateur n'est pas suffisant :

- Un micro-ordinateur a toujours des spécificités.
- Tout logiciel recourt à une ou plusieurs de celles-ci, ceci en fonction ou en s'apparentant à des choix éducatifs.

- La question posée n'est pas génératrice de renseignements didactiques.

Les spécificités du micro-ordinateur ne nous intéressent pas en tant que telles. Ce qui importe, ce sont les spécificités des logiciels en référence à des options didactiques, autrement dit, l'expression en termes didactiques des caractères logiciels.

Il va sans dire qu'une connaissance de l'outil et des logiciels permettant son exploitation est d'une aide certaine, mais la question première doit, selon nous, être d'ordre didactique. Ainsi, le problème ne doit pas être posé par rapport à l'outil, mais par rapport à la didactique de la discipline, en référence à des choix didactiques. Il s'agit de savoir si un logiciel est susceptible d'intervenir utilement, voire même d'apporter un changement dans l'approche des notions, objets d'enseignement, un choix ayant été fait au préalable sur le contenu et l'approche.

Comme les choix didactiques sont décisifs, nous rappellerons brièvement les nôtres (nous les avons déjà évoqués en page 114), car c'est par rapport à ces choix que nous allons analyser ensuite les spécificités des logiciels.

Nous travaillons dans le cadre de l'institution avec les moyens matériels et humains mis à notre disposition.

Nous souhaitons pour les enfants l'acquisition des concepts, ce qui signifie que les élèves sachent les faire fonctionner dans des situations variées. Par conséquent, nos objectifs concernent la conception de problèmes aptes à permettre à l'enfant la construction des savoirs par l'action (en référence aux théories de Piaget), en particulier dans un fonctionnement de type dialectique outil-objet. Nous ne confondons par pour autant action et activisme sans objet d'enseignement. Notamment, nous pensons que la construction des notions nécessite diverses phases, dont celles de formulation, validation, institutionnalisation, phases pendant lesquelles le concours des élèves et de l'enseignant reste indispensable.

⇒ MISE AU POINT : OBJET DU CHAPITRE

Les préalables que nous venons de développer nous permettent de mieux cerner l'objet du chapitre en cours.

Nous allons exprimer des spécificités des logiciels, et cette étude sera menée en accord avec notre position sur l'enseignement.

Ceci implique notamment que nous ne considérons pas qu'un logiciel (ou un ensemble de logiciels) puisse se suffire à lui-même pour l'acquisition d'une notion ; il en est de même de l'étude d'un logiciel par rapport à l'étude de l'acquisition d'une notion. En effet, un logiciel intervient dans une situation, dans une partie de situation ou dans un ensemble de situations, certains moments de l'apprentissage se déroulant sans ordinateur.

Ceci signifie aussi que les spécificités que nous énoncerons nous semblent recevables à un moment ou à un autre des apprentissages : il faut les situer dans le temps de la construction des connaissances.

Nous allons faire une étude fonctionnelle didactique, d'ordre général.

Indépendamment de concepts mathématiques déterminés, nous tenterons de caractériser les logiciels selon des critères didactiques, dans l'objectif de faire apparaître les moments ou les conditions d'usage pertinents. Nous donnerons par là-même un début de réponse à la question formulée lors de la conclusion du chapitre précédent, relativement à la spécificité de la présentation de questions mathématiques par logiciel.

Certains "contextes logiciels" pourront s'avérer originaux, d'autres moins, mais ouvrant toutefois des perspectives intéressantes.

Nous exposerons séparément les diverses spécificités que nous avons pu repérer. Certains logiciels en regroupent plusieurs : est-ce une caractéristique fondamentale de ces logiciels ?

En tout état de cause, que plusieurs aspects figurent dans un logiciel, ou qu'au contraire celui-ci soit très pointu, il peut toujours exister des usages déviants au cours de l'enseignement : ce sujet, bien qu'intéressant, ne sera pas traité.

Ce chapitre comportera une série d'hypothèses dont certaines ont été vérifiées à des degrés divers.

Les logiciels des divers types seront mis en cause.

Nous ne reviendrons pas sur les enjeux que nous avons développés auparavant : nous utiliserons, parfois implicitement, les résultats des chapitres précédents dans notre propos.

Rappelons enfin que nous traitons essentiellement des logiciels à sémantique procédurale.

Plusieurs plans d'exposé étaient possibles.

Nous avons opté pour une présentation des spécificités des logiciels en six sections constituant six modules relativement indépendants.

Ces sections, numérotées de 2 à 7 (la présente étude de la position du problème formant la section 1 du chapitre), ont des contenus qui relèvent de niveaux différents :

- 2 Logiciels pour l'épreuve, la répétition, l'évaluation des savoirs.
- 3 Logiciels pour préparer l'institutionnalisation de notions mathématiques.
- 4 Logiciels pour intervenir sur la gestion de l'activité cognitive.
- 5 Logiciels pour faire respecter des contraintes sur l'expression de la solution au problème posé, ou sur sa méthode de résolution.
- 6 Logiciels pour accompagner l'enseignement ou l'apprentissage.
- 7 Peut-on définir un nouveau cadre : le cadre logiciel, pour des concepts mathématiques.

Ainsi, nous n'avons pas retenu un plan reprenant l'ordre qui peut être globalement celui des diverses phases menant à la construction de connaissances (schématiquement : action, formulation, validation institutionnalisation, épreuve des savoirs).

Par exemple, la section 3 concerne l'institutionnalisation (phase

non initiale de l'apprentissage), et seulement dans une moindre mesure, la construction qui la précède ; celle-ci est plutôt contenue dans les autres sections selon des modalités diverses.

Pratiquement, nous nous sommes intéressée à ce qui nous paraît être des exigences de plus en plus importantes et/ou nombreuses sur les logiciels, des contextes dont l'originalité irait en croissant : l'ordre des sections y correspond.

L'emplacement de la spécificité "Logiciels pour accompagner l'enseignement ou l'apprentissage" en section 6 peut s'expliquer aussi autrement : il s'agit de contextes logiciels adressables non seulement à l'élève, mais aussi à l'enseignant pour une utilisation avec la classe entière.

Il resterait à affiner les résultats et à expérimenter. En particulier, il nous paraît indispensable de trouver des domaines d'application des spécificités que nous allons évoquer, c'est-à-dire, des notions mathématiques, des logiciels et des situations qui les fassent intervenir à fin d'enseignement, permettant de déterminer dans quelle mesure nos analyses a priori des spécificités se retrouvent dans un fonctionnement en situation didactique.

2- LOGICIELS POUR L'ÉPREUVE, LA RÉPÉTITION, L'ÉVALUATION DES SAVOIRS

2a Types de logiciels concernés

Nous ferons une hypothèse sur les logiciels qui apparaissent actuellement caractéristiques de ces spécificités :

Les logiciels de type A, c'est-à-dire à communication I, sont utilisables essentiellement à fin d'épreuve, de répétition, d'évaluation de savoirs, et ceci dans un rapport individuel de l'élève et du logiciel.

Les logiciels de type A non strict ont d'autres spécificités, tandis que ceux de type A strict n'ont que celles-là.

Les paragraphes qui vont suivre constitueront une argumentation en faveur de notre hypothèse.

Pour justifier déjà de façon générale l'énoncé pour sa première partie, nous rappellerons quelques caractéristiques des logiciels A :

Les logiciels à communication I sont des logiciels à mode réponse. Pour satisfaire à une instruction question du logiciel, l'élève ne peut effectuer qu'une seule entrée (éventuellement renouvelable, si l'instruction question est reconduite par le logiciel en cas d'erreur). L'intervention de l'élève consiste donc, soit en une absence de production, soit en une production d'emblée exacte ou inexacte : nous dirons qu'à la production de l'élève est associé l'un des termes du triplet (rien, vrai, faux).

En outre, aucun moyen logiciel ne permet la recherche ou la construction de la réponse par l'élève.

L'évaluation et la correction de la réponse de l'élève sont faites le plus souvent par le logiciel.

Par conséquent, le "contexte logiciel" est compatible avec les spécificités mentionnées par notre hypothèse.

Plus précisément, une ou plusieurs de celles-ci peuvent être présentes dans un même logiciel. Souvent, les fonctions de répétition et d'évaluation coexistent.

On peut se demander quelle est l'influence du genre d'évaluation sur l'existence ou la prédominance d'une des spécificités : en particulier, si le logiciel dispense la réponse exacte, la fonction de répétition ne peut-elle être favorisée? Si le logiciel se contente d'infirmier ou de confirmer la réponse, son rôle n'est-il pas plutôt d'évaluation? Enfin, s'il ne transmet à l'élève aucun élément d'information directement relatif au contenu de sa réponse, des perspectives ne sont-elles pas offertes en matière d'épreuve ayant un caractère différent de celui d'une stricte répétition?

Signalons enfin que lorsque plusieurs spécificités sont potentielles, les conditions d'usage, peuvent décider de la fonction effective. Notamment, la culture de l'utilisateur peut s'avérer un paramètre déterminant : nous pensons même qu'au cours d'un cursus peut avoir lieu un changement de fonction (par exemple, de répétition à évaluation).

2b Types de situations mises en place

Nous rappellerons maintenant qu'un logiciel de type A n'autorise pas de réponses développées, et demande généralement à l'utilisateur de se conformer à un format de réponse.

Ceci nous semble contribuer, avec les précédentes caractéristiques évoquées, au fait qu'un logiciel de type A, et surtout de type A strict, ne propose pas, ou ne peut pas servir à la définition de situation-problèmes.

En type A non-strict, les perspectives en ce domaine semblent plus ouvertes :

Certains logiciels A non-stricts peuvent permettre l'élaboration de situation-problèmes : nous citerons RECT et TACHE pour l'emploi qui en a été fait au cours de la recherche menée par D. Butten. Toutefois, nous n'oublierons pas que ces logiciels s'éloignent du type A selon plusieurs critères (évaluation, mode commande pour objet intermédiaire, existence d'un jeu de cadres). D'autre part, nous soulignerons que ces logiciels n'offrent pas de moyens internes de traiter la situation-problème numérique qu'ils ménagent.

Ainsi, avec un logiciel A, dans la plupart des cas, les spécificités de répétition, d'évaluation, et d'épreuve ne vont pas s'exercer au cours de situation-problèmes, mais pour des questions limitées.

Nous anticiperons sur la suite en signalant que les logiciels A ne sont pas les seuls autorisant notamment l'épreuve de savoirs : d'autres spécificités attachées à d'autres types de logiciels auront les mêmes conséquences, ou une partie d'entre elles. Cela signifie qu'il existe des logiciels qui peuvent combler la carence présentée par les logiciels de type A, en matière de situations-problèmes construites à fin d'épreuve, ou d'évaluation.

2c Types d'intervention dans l'enseignement

Deux types d'interventions nous semblent possibles :

⇒ ACQUISITION DE SAVOIRS NOUVEAUX PAR "MECANIQUE OBJET-OUTIL"

Nous reprenons entre guillemets la terminologie de R. Douady.

Dans la mesure où nous retenons l'hypothèse de R. Douady citée ci-dessous, il ne nous paraît pas déraisonnable, compte-tenu de ce qui a été développé, que des logiciels A interviennent de cette façon.

"Il y a une masse critique de connaissances à acquérir par le processus décrit en a). [Il s'agit de la dialectique outil-objet] - grâce à quoi il est possible de leur intégrer d'autres connaissances acquises autrement, par exemple par la mécanique objet-outil, et qui fonctionneront efficacement. - en dessous de laquelle le "j'apprends j'applique" est inefficace pour la plupart des élèves."
 ("Rapport enseignement apprentissage : dialectique outil-objet, jeux de cadre." R. Douady ; Cahier de Didactique des mathématiques n°3 ; Publication IREM Paris VII).
 Dans ce texte, "mécanique objet-outil" s'oppose à "dialectique outil-objet", et est en quelque sorte synonyme de "j'apprends, j'applique".

Toutefois, selon les choix didactiques en fonction desquels nous conduisons notre analyse, et en accord avec l'hypothèse précédemment citée, de tels logiciels ne nous semblent pas permettre la construction complète de notions. La production de réponses exactes relatives au concept impliqué ("vrai" du triplet (rien, vrai, faux)) proviendrait d'une institutionnalisation par répétition, l'action effective faisant défaut ; en outre, les acquisitions seraient réduites aux termes du logiciel, le statut social d'un logiciel A étant l'utilisation individuelle. Il nous semble repérer là, des apprentissages au sens d'une théorie trop behavioriste.

⇒ INTERROGATIONS SUR DU SAVOIR ANCIEN

Pour répondre aux questions, l'enfant peut mobiliser des savoirs anciens, effectuer une reconstruction d'acquis préalables, extérieure au logiciel (mentale, ou le cas échéant sur papier, ..., un logiciel A n'offrant pas de moyens internes de ce point de vue).

Ainsi, l'une des trois valeurs (rien, vrai, faux) peut être obtenue si des acquisitions préexistent, et/ou si une construction antérieure est mémorisée.

C'est pourquoi, si nous considérons un logiciel A inopérant pour la construction complète de savoirs nouveaux, nous pensons qu'il autorise l'interrogation des savoirs existants relatifs à une notion, et par conséquent

leur renforcement.

Cette consolidation peut intervenir par des questions limitées, selon les trois modes que nous distinguerons de façon plus détaillée par la suite (répétition, épreuve, évaluation).

Enfin, le statut social du logiciel est une utilisation individuelle (dans un contexte, rappelons-le, de jeu contre l'ordinateur). Cela est peu favorable à la construction des fonctions de la communication dans le développement de connaissances.

⇒ MISE AU POINT

Les deux types d'intervention des logiciels A dans l'enseignement que nous venons de déterminer, indiquent que le recours aux logiciels A se situe globalement après un certain apprentissage préalable des notions en jeu, dans un cadre qui comporte autre chose que des logiciels de ce type, présentant ces spécificités.

Des logiciels de type A non-strict dont nous reparlerons, peuvent toutefois déroger à cette règle.

2d Domaines de savoirs concernés

Un logiciel de type A ne permet pas de développement de réponses, ni de construction de celles-ci par l'élève avec des moyens internes au logiciel.

Par conséquent, il nous semble que le domaine de savoirs privilégié des logiciels A est celui des savoirs de types algorithmique ou terminologique.

Néanmoins, nous proposerons au paragraphe suivant au moins un exemple montrant que d'autres domaines peuvent être concernés de façon pertinente, mais ceci par des logiciels A non-stricts.

2e Pertinence des logiciels pour l'épreuve, la répétition, l'évaluation de savoirs

En introduction, nous redirons d'abord les possibilités offertes par les logiciels A en matière d'activités individuelles avec correction immédiate.

Nous continuerons en nous souvenant du désir, observé avec l'utilisation de LOGO, qui existe souvent chez l'apprenant d'une gestion privée de l'erreur entre lui et l'ordinateur : les logiciels A garantissent en général cette condition.

⇒ SPECIFICITE : EPREUVE OU REPETITION

Nous distinguons répétition d'épreuve, le terme épreuve ne signifiant pas pour nous nécessairement répétition.

Nous proposerons un exemple pour différencier épreuve et

répétition :

Si l'on veut s'assurer de la maîtrise par l'élève d'une technique opératoire, on peut lui soumettre diverses opérations : il sera en condition de répétition du savoir en cause.

On peut aussi lui proposer un ou plusieurs problèmes admettant ou non pour modèle l'opération mathématique en question : il sera alors en condition d'épreuve de la technique dans un contexte plus général d'épreuve du sens de l'opération ; il ne s'agira pas d'une simple répétition.

- Répétition stricte

Les logiciels de type A strict qui proposent de nombreux exemplaires de questions isomorphes répondent particulièrement à cette spécificité.

Encore faut-il être conscient des limites existantes : limites indépendantes du logiciel (temps effectif d'utilisation par l'élève), ou intrinsèques au logiciel ou au matériel (taille de l'écran, capacité mémoire ...).

Par ailleurs, nous dirons que la spécificité de répétition nous semble plus directement présente lorsque l'évaluation comporte un affichage des corrigés des questions posées.

Notre propos relatif aux types d'intervention de ces logiciels dans l'enseignement montre que, pour nous, employer un logiciel en répétition, n'est pas synonyme d'emploi en soutien des élèves en difficulté.

La mémorisation, bien que celle-ci ne puisse être fondée sur la seule répétition, peut être l'enjeu de l'usage d'un logiciel de ce type. Les savoirs algorithmiques qui nécessitent consécutivement et parallèlement à d'autres formes d'appropriations, une certaine mécanisation, peuvent faire l'objet d'un traitement par logiciel de répétition : la mémorisation de tables d'opération, de techniques opératoires, de convention d'écritures pourrait être assistée de logiciels répéteurs.

Remarque :

Le logiciel RECT est de type A non-strict.

Nous dirons qu'il pourrait avoir statut de logiciel de répétition, d'épreuve ou d'évaluation s'il affichait une évaluation des réponses de l'élève, avec corrigé.

Dans ces conditions, son emploi à fin d'épreuve de savoirs existants au sujet de la désignation multiplicative pourrait se justifier auprès d'enfants ayant déjà réalisés des acquisitions.

Dans le cas contraire, c'est-à-dire auprès d'élèves n'ayant aucune connaissance relative à cette notation, l'apprentissage ne pourrait relever que d'une mécanique objet-outil qui nous paraît ici formelle et sans support conceptuel préalable.

C'est pourquoi, la conception actuelle du logiciel nous semble bien adaptée au contexte d'usage, c'est-à-dire comme nous le verrons plus loin, en préalable à l'institutionnalisation de la notation multiplicative : en l'absence des corrigés de réponses, on ne pouvait s'attendre à un apprentissage par mécanique objet-outil ; par contre, ce logiciel de type A, non-strict soulignons-le,

présente ainsi une originalité : dans le contexte d'utilisation, celle d'une mise à l'épreuve de savoirs, ceux-ci n'étant pas directement le savoir visé, à fin de constat de leur inadaptation à la situation proposée.

Nous reparlerons de cette originalité qui servira à expliciter une autre spécificité des logiciels.

- Répétition et/ou épreuve

Nous pensons que d'autres domaines mathématiques que ceux des savoirs algorithmiques peuvent être concernés.

Pour ceux-ci, nous dirons tout d'abord que l'épreuve ou la répétition par QCM, si sophistiqué ce dernier soit-il, nous paraît peu efficace en raison du hasard qui peut accompagner les réponses à ce genre de questionnement, et de l'ambiguïté du statut des "réponses" du logiciel.

Par contre, s'il s'agit d'épreuve et non de simple répétition, il pourrait être possible de concevoir des logiciels restant dans le type A, qui ne présentent pas la caractéristique d'isomorphie des problèmes, ou qui développent une forme particulière d'évaluation.

Celle-ci est à définir : par exemple, il pourrait ne pas y avoir de corrigé des questions abordées, ou même pas d'évaluation du tout. Ils s'agiraient alors de logiciels A non-stricts, puisque ne vérifiant pas tous les critères du type I de CONT.

Nous citerons ici en exemple EVAMA (type A non-strict) :

Ce logiciel permet en effet une épreuve des savoirs relatifs au champ conceptuel de la proportionnalité : les problèmes n'y sont pas isomorphes et couvrent un ensemble de types de situations de proportionnalité.

L'évaluation est faite par ailleurs grâce à un bilan exprimé sous forme de conseils concernant les types d'activités à entreprendre. Dans un tel contexte, l'apprentissage par mécanique objet-outil est improbable ; ce logiciel de type A non-strict est ainsi d'une originalité certaine.

Enfin, nous n'oublierons pas qu'avec ces logiciels A, même non-stricts, l'épreuve de savoirs existants intervient dans un fonctionnement en succession de questions/réponses : il existe d'autres formes d'épreuves dans d'autres contextes logiciels, avec d'autres spécificités.

⇒ SPECIFICITE : EVALUATION

Les logiciels de type A permettent l'évaluation de connaissances, dans des limites déjà précisées : notamment, lorsque l'expression des connaissances peut s'effectuer sans développement en suivant un certain "format".

Par conséquent, dans une évaluation par ce type de logiciels sont déterminants le contenu et la formulation des questions.

Une bonne maîtrise de la communication de type I par l'élève est également indispensable.

Deux cas peuvent se présenter :

– L'évaluation faite par le logiciel est transmise à l'élève.

Le plus souvent, il s'agit d'un affichage infirmant, confirmant et/ou signalant la réponse exacte.

Ce contexte respecte le désir de l'élève, dont nous avons rappelé plus haut l'existence possible, que l'erreur reste une affaire privée entre lui et la machine, c'est-à-dire, qu'il n'y ait aucun tiers dans la relation d'évaluation.

Notons que l'émission de la bonne réponse à chaque question par le logiciel ne nous paraît pas nécessaire à la fonction d'évaluation ; peut-être, son absence affirmerait-elle la spécificité du logiciel ?

D'autres formes d'évaluation pourraient aussi être développées, dont celles qui laisseraient à l'élève le soin d'apprécier son activité grâce à un affichage spécifique.

Enfin, il serait souhaitable de compléter les évaluations fragmentaires dues aux logiciels A par d'autres se déroulant avec des moyens informatiques ou non.

– L'évaluation faite par le logiciel n'est disponible que pour l'enseignant.

Le logiciel comporte alors au moins une prise sur fichier des actes de l'élève.

On peut souligner les perspectives offertes sur le plan didactique.

Toutefois, nous ne pensons pas que de tels logiciels soient dans la pratique utilisables en dehors de contextes expérimentaux : en effet, en l'absence d'un logiciel d'analyse du fichier constitué, l'enseignant ne disposera vraisemblablement pas du temps nécessaire au dépouillement.

2f En guise de conclusion

Il est possible de définir une prospective de construction de logiciels à communication I, apparentables au type A, dont la spécificité serait essentiellement la répétition, l'épreuve, l'évaluation de savoirs existants relatifs à une notion, voire leur acquisition dans le cadre d'une mécanique objet-outil.

Aujourd'hui la majorité des logiciels répondant à cette prospective sont de type A strict, c'est-à-dire qu'ils vérifient quasiment tous les critères du

type I de CONT.

Or, nous ne pensons pas que cette identité des logiciels soit une nécessité.

En particulier, la répétition, l'évaluation et l'épreuve ne sont pas des fonctions assimilables.

C'est pourquoi, le développement de logiciels de type A non-strict pourrait constituer une solution. Si les auteurs de ces logiciels tiennent compte de suggestions que nous avons faites pour prendre en compte ces fonctions multiples, l'expérimentation serait alors envisageable.

Il est également possible de projeter des logiciels à communication II, proches des types B ou C, tout spécialement pour l'épreuve de savoirs.

Ceci apparaîtra comme conséquence de l'étude réalisée à la section 7 de ce chapitre.

3- LOGICIELS POUR PREPARER L'INSTITUTIONNALISATION DE NOTIONS MATHÉMATIQUES

3a Types de logiciels concernés

Nous ferons une hypothèse précisant les logiciels actuellement concernés par cette spécificité :

Des logiciels de types A non-strict, B et C sont utilisables pour la préparation de l'institutionnalisation de notions mathématiques ; toutefois, cette préparation revêt un caractère différent selon le type, A non-strict, B ou C du logiciel.

Nous développerons dans les paragraphes suivants une argumentation de cette hypothèse. Le plan adopté vise à mettre en évidence les différences entre le type A non-strict et les types B ou C pour la spécificité étudiée.

C'est pourquoi, nous ferons tout d'abord une remarque générale, indépendante du type de logiciels : puisqu'il s'agit d'une préparation à l'institutionnalisation, les logiciels concernés participent à l'élaboration de situations se situant dans les premières phases de l'enseignement relatif aux objets dont on cherche à effectuer l'institutionnalisation ; en outre, dans notre propos, l'institutionnalisation à proprement parler est faite par l'enseignant.

3b Les logiciels de type A non-strict pour préparer l'institutionnalisation de notions

⇒ PRELABLE : RAPPORT DES ELEVES AU SAVOIR VISE

Compte-tenu de la spécificité étudiée, nous nous intéressons à un usage de ces logiciels auprès d'élèves n'ayant pas les connaissances nécessaires pour répondre aux questions posées, mais celles suffisantes pour les comprendre.

Le contexte suivant peut alors être effectif :

- Le logiciel est de type A, c'est-à-dire à communication I ; par conséquent, il pose un problème à l'élève, mais ne lui procure pas de moyens internes (absence d'instructions de production) pour approcher la solution.

- Si en outre, le logiciel n'intervient pas en donnant indications ou correction, la résolution de la question n'étant pas possible avec des savoirs anciens, l'élève peut s'abstenir de répondre, ou produire une réponse inexacte (il ne sait pas qu'il ne sait pas) et être informé ultérieurement par l'enseignant de l'inexactitude.

Une telle situation peut paraître a priori inadaptée à l'apprentissage, puisque génératrice d'échec.

Nous pensons pour notre part qu'elle est certes difficile à gérer ; mais, si le logiciel impliqué a des caractéristiques supplémentaires que nous allons énumérer, elle peut être pertinente : le logiciel constitue alors un outil didactique intéressant.

⇔ LOGICIELS A NON-STRICT COMME OUTILS DIDACTIQUES POUR PREPARER L'INSTITUTIONNALISATION DE NOTIONS

Nous nous intéressons aux caractéristiques qui peuvent permettre à un logiciel A non-strict d'être un outil didactique susceptible de préparer l'institutionnalisation de notions.

Nous proposerons les caractéristiques suivantes :

- En ce qui concerne le critère "évaluation"

Le logiciel A se borne à infirmer/confirmer les réponses, ou n'évalue pas.

Rappelons que dans le cas contraire, avec une correction systématique, dans le contexte de référence, le logiciel intervient pour un apprentissage relevant de la mécanique objet-outil.

- En ce qui concerne le critère "mobilisation des savoirs de l'élève"

L'objet du problème posé par le logiciel A est dans le champ de connaissances de l'élève ; ce dernier peut envisager ce qu'est une réponse.

Le logiciel autorise une réelle mobilisation de savoirs et de stratégies ; il permet ainsi l'épreuve de leur caractère non-pertinent pour le problème posé.

Par exemple :

- Le logiciel peut imposer une gestion spécifique du temps.
- Le recours à une communication de type II pour la réalisation d'un objet intermédiaire peut aussi s'avérer nécessaire à la mobilisation des savoirs de l'élève.

Nous renvoyons à la section 4 de ce chapitre où figure l'exposé concernant diverses façons pour un logiciel de susciter la mobilisation de savoirs et de stratégies.

Ainsi, le logiciel s'appuie effectivement sur le défaut objectif de connaissances, mais pour convaincre l'élève de la nécessité de savoirs nouveaux.

- En ce qui concerne le critère "approche implicite des savoirs à acquérir"

Le logiciel A permet à l'élève une approche implicite des savoirs à acquérir.

Il n'est pas pour autant possible à celui-ci de les construire avec ce dont il dispose : le logiciel fait manipuler à l'élève des objets pertinents,

compte-tenu du savoir visé, toutefois il ne lui offre pas les moyens de les coordonner seul.

Il nous est difficile ici de préciser le contenu de cette caractéristique au niveau d'un logiciel, de façon générale. Toutefois, nous dirons que l'approche implicite peut se dérouler en communication de type I (c'est-à-dire, à mode réponse) ou encore en communication de type II (avec objet intermédiaire produit en mode commande).

Cette dernière communication a de plus fortes chances d'être nécessaire, si par ailleurs, le logiciel demande une approche implicite du savoir dans un cadre graphique ou géométrique.

⇒ TYPES DE SITUATIONS ET D'INTERVENTION DANS L'ENSEIGNEMENT

Nous dirons qu'il y a création d'un contexte de remise en cause des savoirs existants pour leur pertinence dans le problème, d'explicitation du défaut objectif de connaissance adaptées.

Le logiciel pointe l'échec, tout en offrant une entrée dans le savoir en cause.

C'est pourquoi, nous pensons que si la spécificité en jeu est présente, la situation proposée est une situation problème ; un nouvel objet est en cause, si bien sûr, la question est correctement posée, et le logiciel permet de façon restreinte un fonctionnement de la dialectique outil-objet.

En effet, des éléments de la dialectique outil-objet sont effectifs :

Le problème a un sens pour l'élève. Celui-ci peut exercer ses savoirs dans différentes stratégies pour produire la réponse qu'il envisage. Ce faisant, il rencontre des difficultés dans la résolution parce que ses moyens ne sont pas adaptés.

Toutefois, d'autres éléments de la dialectique outil-objet nous semblent faire défaut :

Dans le contexte que nous avons décrit, les actes de l'élève ont un rôle de révélateur ; il y a bien manipulation de composantes relatives au savoir visé, mais pas de construction de nouveau implicite susceptible d'être formulé et identifié en terme d'objet. Il n'est d'ailleurs pas impossible que ceci soit une conséquence du fait que le logiciel n'offre pas de moyens internes pour traiter le problème.

Par conséquent, nous résumerons en disant qu'un logiciel A non-strict présentant les caractéristiques mentionnées n'a pas les propriétés permettant la construction de savoirs.

Cependant, il propose dans le cadre d'une utilisation généralement individuelle, une situation problème créant les conditions de l'introduction (et

de l'institutionnalisation ultérieure) du savoir par l'enseignant, sur la base d'un vide apparent.

Par vide apparent, nous entendons que l'objet demandé à l'élève (solution du problème, objet de savoir) n'est pas produit. Néanmoins, des acquisitions ont eu lieu : il y a approche implicite d'éléments fondamentaux à la compréhension de l'introduction faite ensuite par le maître, et épreuve de la nécessité du savoir qui sera présenté.

Enfin, pour qu'un logiciel de type A soit pertinent pour la spécificité en cause, il importe que ce logiciel regroupe l'ensemble des caractéristiques citées. La question est ouverte sur le fait que la liste de ces caractéristiques soit exhaustive.

Nous citerons comme exemple le logiciel RECT qui a été conçu par D. Butten et nous-même, à l'époque avec un cahier des charges plus implicite que ce qui précède...

En aucun cas, RECT ne permet à l'élève qui ignore la notation multiplicative de la découvrir. Il ne permet pas la construction implicite de l'objet nouveau de savoir : en particulier, aucun élève ne connaissant pas par ailleurs la désignation axb n'a exprimé le dénombrement des grilles en référence au nombre de points d'une ligne et d'une colonne.

Par contre, RECT place l'élève face à une situation qui a un sens pour lui (dénombrer une collection). Il lui impose une mobilisation de ses savoirs en limitant le temps de réponse et en proposant des grilles de points de tailles convenablement choisies ; ce faisant, il déstabilise les savoirs, en souligne d'autant mieux l'inadaptation qu'il permet à l'enfant de ralentir la machine au moment de l'affichage des grilles. Il fait aussi manipuler les éléments majeurs de la notion à introduire dans un jeu de cadres numérique/graphique : en particulier, il permet une exploration intellectuelle des grilles par un jeu de commandes (en communication II), et donc une approche de l'expression graphique de la notion (nous avons vu qu'ultérieurement, il n'y a pas de compte double ou d'oubli du carreau du coin des grilles). Enfin, RECT n'évalue pas.

La production numérique réclamée est inexistante. Toutefois, l'échec n'est qu'apparent : l'élève s'est convaincu d'une nécessité et a manipulé dans un cadre les éléments de réponse au problème. Il peut les évoquer, ainsi que ses procédures. C'est pourquoi, le maître peut introduire le savoir et l'institutionnaliser de façon pertinente.

Pour plus de détails, il est possible de consulter "Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", thèse de 3^e cycle, Paris VII, D. Butten.

Une interrogation subsiste au sujet du vécu de l'échec par l'élève. L'enfant comprend-il bien que le défaut de connaissances ne lui est pas imputable, autrement dit qu'il s'agit d'une problématique nouvelle?

Ceci nous amène à penser qu'aussi bien pour limiter un éventuel vécu négatif de l'échec, que pour éviter une trop grande déperdition de la

préparation réalisée grâce à l'outil didactique logiciel, la séquence d'introduction/institutionnalisation doit se dérouler très peu de temps après celle qui requiert le logiciel.

Ce parti a été pris pour l'expérimentation de RECT dans des classes de cours élémentaire.

⇒ DOMAINES DE SAVOIRS CONCERNES

Dans l'exemple que nous avons cité, il s'agit de l'introduction d'une écriture symbolique, et de son sens.

Peut-on imaginer d'autres domaines de savoirs?

Nous laissons la question ouverte.

3c Les logiciels de types B, C pour préparer l'institutionnalisation de notions

Nous procéderons en mettant en valeur les différences pour la spécificité de préparation à l'institutionnalisation de notions entre les logiciels B ou C et les logiciels A.

Ce qui suit nous paraît s'appliquer aux logiciels B et aux logiciels C : la particularité de chaque type n'introduit que des nuances.

⇒ PRELABLE : RAPPORT DES ELEVES AU SAVOIR VISE

En raison de la spécificité étudiée, nous nous intéressons, comme en type A, à l'emploi de logiciels B ou C auprès d'élèves n'ayant pas les connaissances nécessaires pour fournir la réponse à la question posée. En outre, nous nous plaçons dans les deux cas où la question est formulée par le logiciel ou par l'enseignant.

Le contexte suivant est généralement réalisé :

- Le logiciel, étant à communication II, met à disposition de l'élève des moyens internes, à savoir des instructions de production pour approcher la solution.

- L'élève exprime sa représentation du problème en proposant une solution construite avec les instructions de production.

La situation peut, cette fois-ci, paraître très favorable à l'apprentissage.

Toutefois, nous n'oublions pas les réserves que nous avons faites à propos de ces logiciels.

C'est pourquoi, tout en soutenant qu'un logiciel B ou C peut intervenir comme outil didactique pertinent, nous nous employerons dans les lignes qui suivent à mieux cerner les possibilités offertes.

⇒ TYPES DE SITUATIONS ET D'INTERVENTION DANS L'ENSEIGNEMENT

- La situation proposée est une situation-problème, à condition bien sûr que la question soumise à l'élève implique au moins l'objet nouveau de savoir dont on veut effectuer la construction et l'institutionnalisation.

- Le fonctionnement de la dialectique outil-objet est possible.

La situation proposée devrait se trouver dans le champ de connaissances de l'élève.

A ce sujet, une première mise en garde s'impose : suivant celui qui définit le problème, il se peut que cette condition ne soit pas satisfaite ; en particulier, si l'élève travaille par projet, il est peu vraisemblable, surtout en type C, que la spécificité étudiée puisse s'exercer.

Un logiciel de type C offre généralement de nombreuses possibilités quant aux concepts et aux problèmes que l'on peut définir relativement à ces concepts. Dans ces conditions, il est improbable qu'un élève puisse se donner un projet spécialement adapté à la notion dont on espère que le logiciel aide à la construction, et prépare l'institutionnalisation.

Ceci explique pourquoi nous ne retenons qu'un usage de ces logiciels pour des questions émises par l'enseignant ou le logiciel lui-même.

Nous verrons en détail à la section 7 de ce chapitre comment les instructions de production fondent la mise à l'épreuve de savoirs, et aussi une production susceptible de contenir implicitement l'objet nouveau de savoir.

Dès maintenant, nous dirons qu'une différence avec les logiciels A peut être là soulignée : avec un logiciel B ou C, il y a presque toujours production, car il y a presque toujours emploi des instructions de production (sauf blocage patent de l'élève ...).

Dans ces conditions, la production peut comprendre des expressions qui pourront être identifiées par l'enseignant dans la construction du savoir, et servir à son institutionnalisation.

Toutefois, la production n'est pas synonyme de construction de savoirs, ni d'absence d'échec.

Nous avons établi des limites de l'usage d'un tel logiciel à fin d'apprentissage mathématique (statut et validation de l'objet). En particulier, l'erreur/l'échec peut passer inaperçu de l'élève (rappelons le carré aux angles de 89° en LOGO).

A l'inverse de ce qui se passe avec un logiciel à communication I, un logiciel à communication II ne propose généralement pas pour un problème donné, l'alternative entre défaut de réponse et réponse inexacte, mais entre réponse inexacte et exacte (ceci, tout spécialement dans le cas du type C). L'appréciation ne peut être laissée au seul élève.

C'est pourquoi, nous soulignerons à nouveau le rôle fondamental de l'enseignant.

Lors d'un usage individuel ou par petit groupe, le logiciel prépare l'élève à une institutionnalisation. Celle-ci ne s'effectue pas sur un vide apparent comme en type A non-strict ; elle intervient sur une production incluant les éléments à institutionnaliser, mais pouvant éventuellement être fautive, voire ne pas avoir initialement de statut mathématique.

⇒ MISE AU POINT

Avec un logiciel de type A non-strict, une dialectique outil-objet restreinte peut fonctionner pour permettre la remise en cause des savoirs existants, et une institutionnalisation succédant à une introduction préalable par l'enseignant : il n'y a pas de production directement opérante pour le savoir visé.

Avec un logiciel de type B ou C, une dialectique outil-objet complète peut fonctionner pour permettre une institutionnalisation par l'enseignant, l'introduction du savoir étant partie intégrante de l'activité menée avec le logiciel : il y a production significative pour le savoir visé.

Un logiciel de type B ou C n'autorise pas plus qu'un logiciel de type A un apprentissage autonome : il crée, comme un logiciel de type A, mais différemment, des conditions de construction du savoir, et précisément ici, d'institutionnalisation par l'enseignant.

Remarquons enfin que l'institutionnalisation sera fondée quel que soit le type de logiciels auquel on a recours, sur l'activité avec le logiciel, mais aussi sur des formulations, des débats, des validations, l'ensemble de ces données participant à la construction du savoir.

⇒ DOMAINES DE SAVOIRS CONCERNES

A ce jour, nous avons observé que les savoirs impliqués par les logiciels B, et surtout C, sont plutôt géométriques pour l'enseignement élémentaire sur lequel nous avons centré notre travail et surtout nos exemples.

La question est là aussi ouverte.

3d En guise de conclusion

Pour la spécificité de préparation à l'institutionnalisation de notions mathématiques par l'enseignant, il est possible de définir une prospective de construction de logiciels participant ainsi à la construction du savoir.

On peut envisager des logiciels à communication I, apparentables au type A, mais présentant des caractéristiques particulières autorisant un fonctionnement restreint de la dialectique outil-objet : nous classons ces logiciels en type A non-strict.

On peut aussi projeter des logiciels à communication II, assimilables au type B ou C, qui permettraient un fonctionnement de cette même dialectique outil-objet de façon plus large.

Ainsi, même si le fondement de l'institutionnalisation diffère suivant le type du logiciel préparatoire, les logiciels des trois types peuvent répondre à cette spécificité.

4- LOGICIELS POUR INTERVENIR SUR LA GESTION DE L'ACTIVITE COGNITIVE

4a Position du problème

Nous nous intéressons ici à l'intervention des logiciels sur la gestion de l'activité que l'élève mène avec leur concours.

Precisons ce que nous entendons par intervention sur la gestion de l'activité :
On demande à l'élève de traiter un problème présenté par un logiciel, ou un problème avec un logiciel.
L'élève peut vouloir organiser son activité d'une certaine façon. Cela lui serait possible, si le problème n'impliquait pas un support logiciel : en effet, un logiciel est susceptible de contraindre, de modifier la gestion "ordinaire" de l'activité qu'aurait eue l'élève, c'est-à-dire d'intervenir sur cette gestion.
Nous allons étudier en quoi et par quels moyens.

Nous n'examinerons que des modes d'intervention dans le cadre théorique dans lequel nous nous sommes placée dans ce chapitre.

Autrement dit, dans un contexte d'apprentissage, nous retenons des interventions susceptibles d'engendrer la mobilisation des savoirs, l'évolution des stratégies, voire des comportements ; c'est pourquoi, par exemple, nous ne considérons pas que les logiciels pour la répétition des savoirs interviennent sur la gestion de l'activité.

La pertinence de la question posée par le logiciel (ou l'enseignant) est évidemment déterminante.

Toutefois, notre propos n'est pas une étude cas par cas.

En nous situant à un niveau général, nous pensons qu'une intervention sur la gestion de l'activité suscitant une mobilisation des savoirs ou stratégies, voire leur évolution, doit réaliser une sorte de "provocation de l'élève". Nous indiquerons pour les logiciels des caractéristiques qui, pouvant avoir cette capacité, constituent suivant la situation des formes d'intervention pertinentes.

Notre exposé distinguera les interventions explicites des interventions implicites.

4b Interventions explicites sur la gestion de l'activité

Nous dirons qu'un logiciel intervient explicitement sur la gestion de l'activité, si l'intervention est clairement notifiée par un élément du logiciel.

Dans ces conditions, pour que le logiciel intervienne explicitement sur l'activité, il faut qu'il en définisse au moins partiellement le contenu.

C'est pourquoi, l'hypothèse suivante peut se justifier :

Des logiciels de type A, ou B peuvent intervenir explicitement sur la gestion de l'activité.

Nous distinguerons parmi les interventions explicites celles que nous qualifierons de directes, d'autres que nous nommerons indirectes :

- Une intervention explicite directe soumet directement l'élève à une réglementation de la gestion de son activité.

- Une intervention explicite indirecte peut provoquer une réglementation de la gestion de l'activité par l'élève; elle ne le fait pas elle-même ; il n'est pas certain que l'élève respecte de lui-même le contrat.

⇒ INTERVENTIONS EXPLICITES DIRECTES

• Le temps

Un logiciel peut gérer de façon performante la variable didactique temps, ceci de différentes façons.

Les vécus de l'élève peuvent être diversifiés.

Un premier style d'intervention

Au niveau du logiciel, l'intervention sur la gestion de l'activité par le temps consiste en une limitation de la durée d'une phase déterminée de l'activité.

Différentes phases de l'activité peuvent être concernées :

- La phase d'entrée de réponse seule.

- La phase relative à un ou plusieurs problèmes posés, ceci séparément ou non, en comprenant ou non les entrées de réponses.

- La phase relative à un cursus de plusieurs problèmes.

Des combinaisons restent possibles entre ces diverses éventualités. Les effets produits ne sont pas les mêmes. Ils dépendent notamment de la situation et des connaissances de l'élève. Une étude cas par cas est nécessaire pour une analyse de cet effet de gestion du temps.

Une gestion de ce style de la variable didactique temps faite par logiciel est performante.

Toutefois, l'enseignant pourrait la réaliser sans ordinateur : ce qui différencie dans ce cas de figure une gestion du temps par l'enseignant d'une gestion du temps par un logiciel, ce sont la précision apportée par le logiciel, et

le caractère individuel de la gestion qu'il effectue auprès de l'élève. En effet, rappelons que le statut social dominant d'un logiciel, en particulier de type A, est l'utilisation individuelle.

La question se pose de savoir si l'utilisation individuelle est réellement une obligation, et surtout, dans le cas de la spécificité étudiée, quelles seraient les conséquences d'une utilisation par deux enfants par exemple.

Le logiciel peut proposer à l'élève une visualisation de l'écoulement du temps.

Plusieurs genres de compteurs existent : des compteurs discrets ou continus pour l'appréciation de temps passé ou restant. Nous les avons déjà évoqué en page 180 en étudiant la complexité éventuelle de leur lecture. Nous devons ici signaler qu'il n'est pas impossible que, face à la visualisation permanente d'un compteur, certains élèves rencontrent des difficultés psychologiques se traduisant par un comportement peu favorable à l'apprentissage : réponse au hasard ou incapacité à répondre.

Le logiciel peut aussi n'afficher aucun compteur.

Dans ce cas, on peut se demander combien d'exemplaires du problème proposé sont nécessaires à l'élève pour percevoir la contrainte relative à la durée ? Quelle appréciation en a-t-il ? Lorsque l'élève a assimilé le contrat sur la durée, les effets pervers signalés ci-dessus ne sont pas exclus ; néanmoins, l'absence de compteur visualisable continuellement peut les limiter.

Enfin, la contrainte exercée sur l'élève pendant une phase de l'activité peut l'amener à modifier sa gestion du temps sur des phases non temporisées : raccourcir la durée d'une phase peut le conduire à chercher, — si le logiciel le permet —, à allonger la durée d'autres phases pour reconstruire ailleurs le temps dont il est privé. Ce comportement peut bénéficier à l'ensemble de l'activité.

Nous citerons comme exemple de ce style d'intervention le logiciel RECT :

RECT limite le temps de réponse de l'élève. Celui-ci peut en être prévenu par l'enseignant, par contre aucun compteur ne le renseigne sur le temps écoulé ou restant.

Cette temporisation visait à la mobilisation de procédures efficaces de détermination d'un cardinal, en particulier à l'émergence d'une désignation assimilable à une désignation multiplicative, ou carrément multiplicative.

Seuls les élèves connaissant par ailleurs la notation axb l'ont effectivement mobilisée.

Pour les autres, l'échec était prévisible. Toutefois, en l'absence de disponibilité de l'écriture axb , et compte-tenu des possibilités

offertes dans la phase d'affichage de grille (absence de temporisation et surtout communication de type II pour l'exploration de l'objet graphique grille), on a assisté à des évolutions de stratégies d'affichage visant à ralentir le logiciel pour permettre le dénombrement : ainsi a eu lieu une reprise du temps par l'élève. Nous n'en faisons pas une analyse négative : de multiples stratégies de dénombrements sont intervenues ; même si certaines d'entre elles étaient à un moment donné des régressions par rapport à l'objectif, car plus directement liées au ralentissement du logiciel, l'effet secondaire a été positif : à l'institutionnalisation par l'enseignant, des comparaisons ont pu avoir lieu, le sens et l'efficacité de la désignation axb ont été préparés par l'action. En outre, a vraisemblablement été évité un effet pervers pour les élèves n'ayant aucune connaissance de la notation axb : s'il ne leur avait pas été possible de gérer le temps des phases non temporisées, ils n'auraient vraisemblablement pas répondu du tout ou encore répondu au hasard.

Pour plus de précisions, le lecteur peut consulter "Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire", thèse de 3^e cycle, D. Builen.

Un autre style d'intervention

Nous parlerons maintenant d'une forme d'intervention difficilement réalisable par l'enseignant seul, même de façon imparfaite.

Il est possible de concevoir un logiciel de type A ou B, tel que :

- à chaque question est attachée une durée déterminée ;
- les questions apparaissent à intervalles de temps déterminés.

Si l'intervalle de temps est tel que l'apparition d'une question succède à la réponse à la question précédente, on retrouve le style d'intervention déjà traité (réglementation de la durée de l'ensemble du corpus par celui de chacun de ses éléments).

Par contre, si l'intervalle de temps est tel que des questions s'affichent successivement, plusieurs d'entre elles restent simultanément non résolues, chacune étant par ailleurs temporisée, on est dans un contexte original.

L'élève est confronté à une tâche de gestion de l'ordre de traitement des questions. Si celles-ci ont, du point de vue du contenu, un lien pertinent entre elles, l'évolution des savoirs ou le développement de procédures adaptées peut intervenir en conséquence de la tâche de décision sur l'ordre.

Le discours n'est plus linéaire, on pourrait presque dire que de tels logiciels sont munis de la quatrième dimension "temps".

Nous citerons comme exemple le logiciel de type A non-strict INVASION DES CHIFFRES (Ed. Cedic/Nathan).

Ce logiciel propose un certain type d'activités de calcul mental.

Il fait intervenir la variable didactique temps de la façon décrite

ci-dessus : des opérations envahissent régulièrement et successivement l'écran ; l'élève doit calculer un maximum d'opérations avant que celles-ci ne se détruisent en atteignant le bas de l'écran.

En limitant le temps, le logiciel restitue une condition de vécu et de développement de procédures, propre aux situations de calcul mental.

De plus, nous signalerons que pour l'élève "individuel", la situation seul à seul avec le logiciel introduit une différence notable avec la situation en classe.

En outre, si les opérations sont bien choisies, c'est-à-dire en particulier si, dans un but d'acquisitions relatives au calcul mental, certaines d'entre elles peuvent servir à en calculer d'autres, le contexte peut être spécialement intéressant : l'élève peut tenter de gérer l'ordre de traitement des opérations pour s'aider de résultats intermédiaires.

Il ne semble pas que INVASION DES CHIFFRES respecte dans sa forme actuelle cette clause de lien entre les diverses opérations. Remarquons d'ailleurs qu'un contrôle portant sur le contenu des affichages, s'il n'exclut pas complètement le tirage au hasard des questions, en impose tout au moins la maîtrise.

Nous poursuivrons au paragraphe suivant l'analyse de ce logiciel.

Détourner la limitation du temps

On peut essentiellement au niveau logiciel fixer la durée, donc la limiter.

Toutefois, pour l'élève, selon la situation et ses connaissances, le vécu peut être de l'ordre d'une restriction effective, ou au contraire d'une perte de temps.

Nous poserons le problème de l'intervention d'un logiciel sur la gestion de l'activité par le temps, par rapport à l'élève, ainsi :

Un logiciel peut réglementer le temps sur l'activité à l'inverse de ce que serait la gestion ordinaire du temps par l'élève.

C'est pourquoi, dans l'objectif d'amélioration de l'apprentissage, nous demandons s'il ne pourrait pas être souhaitable dans des situations qui restent à déterminer, d'envisager une gestion du temps par le logiciel qui correspondrait pour l'élève non pas à une réduction du temps imparti, mais au contraire à son allongement.

Nous reprendrons ici l'analyse d'INVASION DES CHIFFRES.

En effet, pour des raisons techniques de départ, ce logiciel fait intervenir le temps autrement que par limitation, à savoir par allongement : la frappe de la réponse doit être faite très lentement pour qu'elle soit prise en compte par le logiciel.

Ceci peut paraître négatif : quand l'élève a trouvé une réponse, il devrait pouvoir la transmettre rapidement : une perte de temps intervient.

Toutefois, nous proposerons en hypothèse cette autre analyse : il n'est pas impossible qu'une fois admis par l'élève, ce

ralentissement l'amène à reconsidérer son comportement sur l'activité : des difficultés psychologiques liées aux restrictions sur la durée peuvent s'atténuer ; l'élève peut prendre le temps de coordonner ses actions (recherche de procédures efficaces, frappe), distinguer précipitation et vitesse, gérer différemment son temps.

La pertinence de cette double intervention du temps par limitation et allongement peut être fonction de l'élève, de l'équilibre entre tâche et temps.

• Accès contrôlé à certaines parties du logiciel

Certaines parties d'un logiciel peuvent se vouloir des aides à la résolution du problème posé.

Nous ne discutons pas ici de leur contenu, et a fortiori de leur pertinence.

L'accès à de telles parties d'un logiciel peut être optionnel, ou encore peut donner lieu à la modification d'un score : dans ce cas, il s'agit d'interventions explicites indirectes dont nous traiterons plus loin. Il peut aussi être réglementé, pour favoriser l'évolution des stratégies ou la mobilisation des savoirs. Cette réglementation peut consister en une interdiction à certaines phases de l'activité, ou après un certain nombre de recours, ou encore pourquoi pas en une obligation d'usage : il s'agit bien alors d'une intervention explicite directe.

Dans le logiciel PUCE, l'emploi de la partie "calculatrice", tout en étant optionnel, n'est autorisé qu'un certain nombre de fois sur une partie déterminée du cursus. Il y a donc intervention explicite directe, nuancée par l'aspect optionnel.

La "calculatrice" peut libérer l'élève de tâches strictement algorithmiques sur une partie du cursus : schématiquement, l'élève commande à la machine les calculs qu'il estime susceptibles de lui permettre la résolution du problème ; il peut fixer sa réflexion sur la recherche du modèle correspondant au problème posé.

En supprimant à un certain stade l'option "calculatrice" à l'élève, l'objectif peut être la mobilisation des acquis implicites de la partie du cursus comprenant la "calculatrice", l'obligation de recourir au calcul mental, pour susciter si possible l'émergence du modèle multiple d'un nombre.

Néanmoins, après observation, on ne peut conclure sur la réelle opérationnalité de ces choix : en effet, ce logiciel, conçu par des enseignants en formation, présente un défaut d'algorithme ; en outre, de type A strict, son fonctionnement est entièrement en mode réponse, même pour le parcours du logiciel et l'option "calculatrice" : ceci provoque une grande rigidité d'emploi, l'utilisateur n'est jamais libre de décider de la partie du logiciel qu'il veut adresser, il doit attendre une question lui offrant un accès.

Dans ces conditions, lors d'une expérimentation, le recours à la "calculatrice" a été systématique, mais celle-ci n'a pas eu la

fonction décrite ci-dessus : l'élève recherchait une forme non canonique du nombre qu'il envisageait comme solution du problème posé par PUCE, la soumettait à la calculatrice pour que celle-ci la lui restitue sous forme canonique.

On peut tenter d'expliquer ce phénomène :

Bien que fonctionnant extérieurement par questions/réponses, la communication de la partie "calculatrice" du logiciel est de type II : peut-être, l'élève satisfaisait-il là son désir de commander l'ordinateur, à moins que, compte-tenu du fonctionnement global à mode réponse très prégnant, il ne se soit pas senti le droit de refuser l'option "calculatrice". Encore lui fallait-il alors trouver une tâche à traiter : toujours en raison du poids de la communication I sur l'ensemble du logiciel, la première qui se présentait à son esprit ou lui semblait cohérente avec le reste de son activité pouvait effectivement être celle qu'il retenait.

Un contrôle d'accès plus cohérent pourrait renseigner de façon plus pertinente au sujet de l'effet sur l'activité cognitive de l'accès contrôlé à certaines parties d'un logiciel.

⇔ INTERVENTIONS EXPLICITES INDIRECTES

Rappelons qu'une intervention explicite indirecte peut conduire l'élève à régler lui-même sa gestion de l'activité ; il n'est toutefois pas certain que ce contrat soit respecté.

• Aides optionnelles

Nous les classons sous la rubrique "interventions explicites indirectes", car au niveau strict du logiciel, rien n'oblige l'élève à les employer.

Toutefois, nous avons vu au cours de l'exemple précédent que le caractère optionnel pouvait être occulté : en particulier, le contexte général du logiciel peut le transformer dans la pratique en caractère de recours obligatoire.

• Présentation sur la même page écran de plusieurs problèmes

Pour que ce genre d'intervention soit pertinent pour intervenir sur la gestion de l'activité, il est vraisemblable qu'il faille que les problèmes concernent le même savoir, certains d'entre eux servant à la résolution des autres. En outre, l'élève doit pouvoir choisir l'ordre de leur traitement.

C'est pourquoi, nous parlons d'intervention explicite (plusieurs problèmes sont imposés), indirecte (l'élève fera-t-il effectivement un choix ? celui-ci sera-t-il pertinent pour la résolution des problèmes concernés ?).

Rappelons que le temps peut intervenir explicitement pour soutenir le contrat.

Par conséquent, INVASION DES CHIFFRES est un exemple de logiciel avec interventions explicites directes et indirectes.

• Score

Nous avons déjà évoqué longuement le score comme message autre que d'évaluation stricte (cf p. 180-183).

Nous résumerons succinctement son intérêt : évaluation formative non explicitée, il peut mener l'élève qui en souhaite l'amélioration, à reconsidérer son activité, à engager des validations partielles, à développer des procédures et à mobiliser des savoirs. Il peut également servir à une recentration sur l'objet du problème.

Nous ne reviendrons pas sur la difficulté didactique à son emploi, sur la difficulté à en apprécier les effets, voire sur l'impact négatif qu'il peut avoir sur l'activité.

• Intervention d'un jeu de cadres

Nous terminerons l'inventaire des interventions explicites indirectes par ce thème qui nous semble en relever jusqu'à un certain point.

Un logiciel de type A, voire B peut proposer une notion dans un jeu de cadres : RECT l'atteste. L'objectif principal est une meilleure appréhension du concept. Nous dirons qu'il y a intervention sur la gestion de l'activité, dans la mesure où, avec une telle présentation, on pense que la mobilisation des savoirs est davantage disponible.

4c Interventions implicites sur la gestion de l'activité

Nous parlons d'intervention implicite lorsque :

- le logiciel n'inclut pas d'éléments spécialement destinés à une réglementation de l'activité ;
- néanmoins, certaines de ses caractéristiques paraissent à un deuxième niveau de lecture, intervenir sur la gestion de l'activité pour la mobilisation ou l'évolution des savoirs, des stratégies.

Nous faisons l'hypothèse suivante :

Des logiciels de types B ou C peuvent intervenir implicitement sur la gestion de l'activité, par la communication de type II qu'ils instaurent.

Nous allons expliciter notre hypothèse, tout en rappelant certains points déjà développés.

Le statut social d'un logiciel à communication II, surtout de type C, est celui d'une utilisation individuelle ou par petits groupes, avec la volonté de faire faire à l'ordinateur ce dont on a décidé (que le problème ait ou non été donné par l'enseignant).

Nous avons dit, à propos de la validation de la composante

mathématique d'une activité avec un logiciel à communication II, que l'élève peut reconsidérer son activité en raison du conflit «machino-cognitif» naissant de la distance entre ce qu'il veut obtenir et ce qui se produit et s'affiche. Il peut alors s'en suivre des pratiques locales de validation, des évolutions de savoirs, de stratégies.

C'est ainsi que nous situons d'une première façon, l'intervention implicite sur la gestion de l'activité de la communication de type II : instauration chez l'élève du besoin de produire ce dont il a décidé (motivation) et conflit «machino-cognitif».

Nous distinguons le conflit «machino-cognitif» du conflit socio-cognitif par une propriété importante :

Un conflit socio-cognitif instaure une situation d'argumentation entre élèves.

Dans le cadre d'un conflit «machino-cognitif», le logiciel se borne à attester l'action de l'enfant par un affichage ; il n'argumente pas, ni l'élève.

Si l'élève admet la fiabilité du logiciel (et de la machine), il remet en cause son activité en cas d'affichage ne lui paraissant pas conforme à son attente : ainsi, toute la tâche relève de l'élève ; au conflit «machino-cognitif» succède un conflit de l'élève avec lui-même.

Nous dirons que ce conflit est notoirement plus délicat à gérer que le conflit socio-cognitif : même si l'élève évite les argumentations fausses de ses pairs, il est exposé à ses propres lacunes ; en l'absence de culture mathématique, le conflit n'existe qu'en cas d'appréciation perceptive négative ; dans le cas contraire, rien ni personne ne remet en question le jugement de l'élève.

Par conséquent, ce conflit est fermé sur l'élève et ses savoirs. C'est pourquoi, si intéressantes que soient les perspectives offertes par la communication II, nous redirons ici qu'elles ne peuvent résoudre les problèmes de l'apprentissage en mathématique.

Remarquons que si deux élèves sont confrontés ensemble à un conflit «machino-cognitif», la situation peut s'évaluer différemment : cela peut ouvrir le débat entre eux sur l'analyse de ce que "fait" la machine par rapport à leur propre objectif.

Un autre aspect de l'intervention de la communication de type II sur lequel nous nous sommes déjà interrogés concerne l'influence de la dialectique apprentissage de la communication/apprentissage mathématique (cf tâches de communication p.139-154).

Les expérimentations sont certainement plus difficiles à mener avec un logiciel que dans un cadre plus classique.

4d En guise de conclusion

Pour la spécificité d'intervention sur la gestion de l'activité, il est possible de définir une prospective de construction de logiciels.

On peut envisager des logiciels à communication I effectuant de telles interventions : celles-ci sont jusqu'à maintenant toujours explicites, qu'elles soient directes ou indirectes.

On peut aussi concevoir pour cette spécificité des logiciels à communication II. Selon nous, ceux-ci interviennent toujours en raison même de l'existence de ce type de communication, c'est-à-dire que leurs interventions sont implicites.

Ainsi, les trois types de logiciels peuvent répondre à une telle prospective.

Des recherches pourraient se développer, avec un objectif de diversification. Il est vraisemblable, et déjà des logiciels de type B ou A non-strict l'attestent, que des interventions multifformes (explicites et implicites) peuvent coexister.

Enfin, quelles perspectives offriront les systèmes multi-tâches? Ou même déjà, que peut-on imaginer sur réseau d'ordinateurs?

5- LOGICIELS POUR FAIRE RESPECTER DES CONTRAINTES SUR L'EXPRESSION DE LA SOLUTION AU PROBLEME POSE, OU SUR LA METHODE DE SA RESOLUTION

5a Position du problème

Il peut être intéressant au cours d'un apprentissage de proposer des problèmes exigeant que l'élève engage des procédures particulières ou exprime de façon spécifique sa solution.

Nous examinerons ici les logiciels du point de vue de celles des caractéristiques qui sont susceptibles de contraindre l'élève en ce sens.

Il ne s'agit pas, comme à la section précédente, d'une intervention sur la gestion de l'activité due à l'apport d'éléments extérieurs au sujet du problème (temps, par exemple), mais d'une imposition portant sur l'expression de la solution ou de la résolution. Cependant, il n'est pas improbable que des éléments d'intervention sur la gestion de l'activité renforcent les éléments de contrainte que nous allons maintenant évoquer.

5b Logiciels pour faire respecter des contraintes relatives à l'expression de la réponse au problème posé

Les contraintes relatives à l'expression de la réponse peuvent consister en une obligation d'emploi de certains caractères, de certains mots, avec le cas échéant, une syntaxe particulière : en bref, il s'agit de contraindre à un "format" déterminé.

A notre connaissance, tous les logiciels effectuent une évaluation syntaxique/sémantique ; par conséquent, ils sont aptes à engager au respect de ce genre de consigne.

C'est pourquoi, nous énoncerons cette hypothèse :

Les logiciels des trois types peuvent faire respecter des contraintes relatives à l'expression de la solution au problème posé.

Remarquons qu'il est évidemment souhaitable que le contexte justifie de telles contraintes, ou que le logiciel ait d'autres caractéristiques rendant son usage pertinent.

En l'état actuel des logiciels, on peut distinguer ceux qui ne pratiquent que cette évaluation syntaxique/sémantique (c'est-à-dire globalement, ceux à communication II), de ceux qui la complètent d'une évaluation relative à l'attendu conceptuel (c'est-à-dire globalement, ceux à communication I).

Rappelons que l'affirmation qui précède est issue de notre analyse de la communication proposée par un logiciel à l'utilisateur : contrairement aux logiciels à type de communication I, c'est-à-dire à mode réponse, les logiciels à type de communication II, c'est-à-dire à mode commande, n'effectuent pas d'évaluation par rapport à l'attendu conceptuel (voir le descriptif de la typologie COM, p. 98-103). Par exemple, LOGO (type II), ne vérifie pas que l'objet produit par l'élève est bien un carré ; nous avons examiné les problèmes posés par la validation de l'activité avec un logiciel à communication II en pages 186-196.

Pour notre part, nous effectuerons ici une différenciation des logiciels, non pas sur le critère du nombre d'évaluations, mais sur celui du type de communication, ce dernier nous semblant permettre de caractériser deux sortes de situations pertinentes pour le genre de contrainte étudiée.

⇔ LOGICIELS A COMMUNICATION I

Les logiciels concernés sont de type A.

L'étude faite des tâches de communication I (cf p. 175-185) a signalé la contrainte exercée par la communication I quant au formatage des réponses, mais aussi l'intérêt d'un formatage faisant partie intégrante de l'activité.

Nous ajouterons que, pour que la spécificité étudiée puisse exister, il n'est pas nécessaire qu'une évaluation par rapport à l'attendu conceptuel accompagne l'évaluation syntaxique/sémantique : de ce point de vue, les logiciels A ont souvent une caractéristique (évaluation par rapport à l'attendu conceptuel) qui n'est pas indispensable.

Nous avons cité le logiciel TACHE qui exige une expression multiplicative, se justifiant à un stade donné de l'apprentissage. Le contrôle relatif à l'attendu conceptuel fait par le logiciel, c'est-à-dire le contrôle du sens de la réponse de l'élève, répond à d'autres objectifs que celui de contrainte d'expression : on pourrait imaginer qu'il y ait seulement le contrôle de la présence de la forme «axb».

Nous ne reviendrons pas sur les problèmes liés à la communication de type I. Nous préciserons seulement, à partir d'un argument déjà développé, les situations appropriées à une exigence de "formatage" de réponse : d'une façon générale, la communication de type I ne permet que difficilement des formulations développées (absence d'instructions de production) : c'est pourquoi, il nous semble que les situations où la spécificité de "formatage" est adaptée sont celles qui ne réclament pas une large expression, en particulier une description, de la part de l'élève ; nous dirons que ce ne sont pas des situations d'expression.

Si d'autres domaines que celui des conventions d'écriture ou des formulations canoniques peuvent être envisagés, cette dernière situation est sans doute l'une de celles où la spécificité en question est la plus pertinente.

⇔ LOGICIELS A COMMUNICATION II

Les logiciels concernés sont de types B et C.

La communication de type II implique des contraintes d'expression : il faut utiliser les termes admis du logiciel, selon sa syntaxe le cas échéant (surtout en type C).

Mais aussi, la communication de type II donne à l'élève, dans le cadre des instructions qu'elle met à sa disposition, toute liberté pour décrire, exprimer sa solution au problème.

C'est pourquoi, nous dirons que la communication de type II autorise l'intervention du logiciel pour des situations de formulation, d'expression.

Nous avons d'ailleurs déjà signalé que, parfois, avec un logiciel à communication II, la tâche de l'élève est principalement de l'ordre de la communication.

Même si d'autres spécificités peuvent lui être reconnu, un logiciel à communication II est selon nous un outil de formulation par excellence (nous reviendrons sur ce point à la section 7 de ce chapitre). Ceci est particulièrement vrai à l'heure actuelle des logiciels de type C (surtout du sous-type 2 II+) : ils permettent, en recourant ou non à un mode différé, de transmettre un texte qui ne sera pas entrecoupé d'instructions de parcours émises par le logiciel (comme cela est le cas en type B).

En résumé :

– les situations ici adaptées sont celles qui réclament une large expression, en particulier une description, de la part de l'élève ; nous dirons que ce sont des situations d'expression.

– Les logiciels les plus adaptés sont ceux de type C.

Certaines situations demandent une formulation rigoureuse, comme élément central de la situation : nous voulons parler des situations de communication au sens ordinaire du terme.

Ces situations ont été étudiées en géométrie par C. Laborde et M. Guillerault.

Dans une telle situation, l'élève (ou un groupe d'élèves) élabore un

message à propos d'un objet, généralement descriptible par un algorithme ; ce message est transmis à un autre élève (ou groupe d'élèves) qui doit produire un objet résultant du décodage qu'il en fait.

Se posent plusieurs problèmes, dont celui de la lisibilité et de la lecture du message, et celui de la comparaison des deux objets.

Nous confirmons ici, qu'une telle situation demandant une rédaction relativement libre d'un texte, les logiciels du sous-type 2 II+ du type C paraissent mieux appropriés à son traitement ; cependant, il n'est pas impossible que les autres logiciels à communication II puissent intervenir s'ils présentent un caractère de généralité adapté à l'objet d'enseignement.

Résumons la situation :

Un émetteur E adresse un message M relatif à un objet O, à un récepteur R. R produit un objet O' correspondant au décodage M' qu'il effectue de M. Les problèmes sont les suivants : M est-il décodable ? M' est-il un décodage de M ? O' correspond-il à M ? O est-il identique à O' ?

Si le récepteur est un logiciel (R=LO) susceptible d'exécuter le message, alors des distorsions de décodage et de production ne peuvent avoir lieu.

En effet :

LO décode exactement M, et produit O' comme décodage de M.

Autrement dit, si R=LO alors M=M' ; en outre, si M respecte les règles de communication régissant LO, alors O' est produit à partir de M, et lui correspond.

Par conséquent, dans une situation avec logiciel, si O diffère de O', le message M est en cause.

Dans une situation sans logiciel, comme des distorsions peuvent s'introduire au moment du décodage M' et de la production O', le message M n'est pas nécessairement en cause.

Signalons toutefois que si la situation avec logiciel apparaît performante à certains égards, elle est aussi une restriction de la situation sans logiciel : aucun élève n'a de tâche de décodage, ni de réalisation correspondant à celui-ci.

Discutons des avantages et inconvénients de cette restriction :

– Tout d'abord, l'élève émetteur E, ayant éprouvé la nécessité d'un message signifiant pour le logiciel, peut être plus directement contraint à produire un message soigné dans son expression, employant des termes permis.

Ceci suppose une maîtrise de la communication proposée par le logiciel. En outre, l'expression de l'élève est assujétie aux systèmes d'instructions. Sont-ils adaptés, suffisants ? La formulation n'emprunte ni un langage naturel,

ni un langage d'ordre mathématique, mais le langage de commande ou de programmation du logiciel. Quelles vont en être les conséquences ? En particulier, comment les éléments signifiants pour l'apprentissage mathématique seront-ils mis en évidence ?

— On pourrait imprudemment penser que, grâce au récepteur logiciel supposé fiable, il est aisé pour l'élève d'apprécier son message M, en appréciant la production O'.

Or, nous savons que la capacité de l'élève à valider une activité avec un logiciel à communication II est limitée : l'élève peut ne pas percevoir une différence existant entre O et O' ; il peut ne pas l'avoir remarqué, ou encore celle-ci peut ne pas exister au niveau perceptif, mais au niveau mathématique.

Par contre, si la distance entre O et M est grande, alors la distance entre O et O' l'est aussi, et dans ce cas, l'élève peut rapidement et sûrement invalider son message.

Remarquons que ceci ne signifie pas pour autant qu'il saura le rectifier efficacement.

En outre, dans une situation avec logiciel, on évite un écueil de situations sans logiciel, relatif à la validation : il est parfaitement possible dans une situation sans logiciel que l'objet O' soit très proche de l'objet O, bien que M s'avère insuffisant.

Autrement dit, contrairement à un récepteur logiciel, un récepteur élève peut avec un message M défaillant produire l'objet O (ou presque), en raison d'un décodage et/ou d'une réalisation ne correspondant pas aux termes de M, mais compatible avec les intentions du rédacteur.

Par exemple, il avait été proposé à des élèves de cycle moyen une situation de communication relative à des polygones, pour une mise en évidence de l'importance des angles dans leur définition. Un des polygones faisait intervenir des angles droits. Bien que dans le message M d'un groupe d'élèves ne mentionne aucun angle, le polygone O' s'est avéré quasiment identique au polygone O. Ceci peut s'expliquer : la production de O' s'est appuyée sur le message M et sur les représentations qu'ont les élèves des figures géométriques en cadre scolaire, c'est-à-dire faisant intervenir des angles droits.

La pertinence de l'objet O proposé par l'enseignant, est évidemment en cause. Toutefois, avec un logiciel, on aurait évité cette ambiguïté ; cela ne prouve pas que des acquisitions auraient eu lieu pour autant ...

— Enfin, les argumentations développées entre élèves dans des situations de communication sans logiciel, lors des conflits dus aux distances entre M et M', entre M' et O', voire à l'incompréhensibilité de M ou à

l'incapacité de R à le comprendre, risquent d'être éliminées : dans une situation de communication avec logiciel, le conflit « machino-cognitif » dont nous avons parlé remplace le conflit socio-cognitif ; toutefois, le travail par petit groupe peut le réintroduire en partie.

En conclusion, des situations de communication avec logiciel nous paraissent intéressantes. Distinctes de celles qui n'en emploient pas, elles nécessiteraient expérimentations comparatives.

5c Logiciels pour faire respecter des contraintes relatives aux procédures à engager pour la résolution du problème posé

Nous nous intéressons maintenant aux situations pour lesquelles l'enseignant veut que soient mobilisées par l'élève certaines procédures qu'il estime plus aptes à provoquer des acquisitions. L'expression de la réponse peut être libre le cas échéant.

Pour qu'un logiciel contraigne l'élève sur ce point, il faut qu'il lui donne le moyen d'exprimer ces procédures et seulement celles-ci.

Par conséquent, nous ferons cette hypothèse :

Des logiciels à communication II, adaptés pour leurs systèmes d'instructions au traitement d'un ensemble réduit de questions, c'est-à-dire actuellement de type B, voire du sous-type 2II- du type C, permettent de contraindre l'élève dans les procédures à retenir pour résoudre un problème.

Il va sans dire que les logiciels de type C autorisent l'expression de procédures. Toutefois, dans la mesure où leurs instructions permettent généralement de traiter des questions très diversifiées, il se peut qu'ils laissent par là-même une trop grande liberté d'approche du problème : ils sont capables d'en interpréter de nombreuses descriptions. Ceci explique la restriction présente dans l'hypothèse ; toutefois, selon le problème, certains logiciels de type C, en particulier du sous-type 2II- pourraient répondre à la spécificité visée.

Le type B de logiciels paraît effectivement bien adapté à limiter les procédures employées par les élèves : les logiciels de type B concernent généralement des domaines plus restreints que ceux de type C en raison de leurs instructions ; leurs systèmes d'instructions sont à indices plus faibles ; en particulier, les productions peuvent avoir lieu par description partielle (paramétrage de l'objet, production à partir de sous-objets et/ou opérations, ...) ; en outre, leurs instructions de parcours peuvent servir à

canaliser la démarche de l'élève.

Nous proposerons comme exemple le logiciel SIMUL (type B) :
Les instructions de SIMUL permettent des analyses par lignes et colonnes de l'objet graphique "grille rectangulaire", mais en interdisent des visions diagonales.
Dans la mesure où ce logiciel est employé pour préparer le sens de la notation multiplicative dans le cadre graphique, il réalise une contrainte pertinente.
Un autre objectif ne ferait pas nécessairement choisir ce logiciel : il pourrait se révéler inadapté en raison même de sa spécificité.

5d En guise de conclusion

Il est possible de définir une prospective de construction de logiciels pour faire respecter des contraintes sur la formulation des réponses à un problème, ou sur la démarche, les procédures.

Des logiciels à communication I peuvent utilement autoriser une contrainte sur le format de réponses ne relevant pas d'une description.

Des logiciels à communication II répondent diversement à la prospective :

Actuellement :

– des logiciels du sous-type 2II+ du type C, (voire du sous-type 2II-) sont plus aptes à contraindre de façon pertinente sur la formulation de réponses relevant d'une description. En effet, ils acceptent des textes pour les exécuter, le cas échéant de manière différée ; ils n'interrompent pas l'utilisateur par des instructions de parcours.

– des logiciels de type B, (voire du sous-type 2II- du type C) sont plus aptes à restreindre les procédures employées par l'élève. En effet, leurs systèmes d'instruction ont un moins grand caractère de généralité ; en type B, les procédures peuvent même être organisées et la démarche de l'élève canalisée par les instructions de parcours.

On pourrait s'intéresser au développement de tels logiciels dans divers domaines mathématiques, mais aussi au développement de logiciels analogues à ceux du type C pour la capacité d'accueil de textes, et analogues à ceux de type B en ce qui concerne la moins grande généralité de leurs instructions : il s'agirait de mini-interpréteurs spécialisés.

6- LOGICIELS POUR ACCOMPAGNER L'ENSEIGNEMENT OU L'APPRENTISSAGE

6a Position du problème

Certaines tâches de l'enseignant ou de l'élève peuvent être facilitées par l'emploi d'un logiciel. Ceci nous paraît pouvoir influencer sur l'enseignement ou l'apprentissage.

Nous proposerons une liste non exhaustive de fonctions d'accompagnement, aussi bien de l'apprentissage que de l'enseignement : dans la plupart des cas, seule la mise en œuvre précisera le rôle rempli.

6b Logiciels pour illustrer

Nous nous intéressons aux logiciels permettant l'illustration de questions mathématiques, en géométrie (images graphiques, ...) ou dans le domaine numérique (images numériques, ...).

L'enseignant peut recourir à un tel logiciel pour préparer, argumenter, renforcer l'enseignement qu'il fait d'une notion : à certaines étapes, il intervient auprès de la classe avec un logiciel.

L'élève (ou un petit groupe d'élèves) peut aussi travailler de façon individuelle avec un logiciel d'illustration : il explore, peut percevoir des éléments caractéristiques de la notion, émettre des conjectures. En quelque sorte, il pratique une mathématique expérimentale ; le logiciel s'apparente dans ces conditions à ce que certains nomment «logiciel de simulation».

L'illustration d'une question sera acceptable si elle présente au moins les qualités suivantes : obtention à la demande, simple et rapide ; autorisation d'affichage de nombreux exemplaires correspondant à des interrogations multiples, voire à des approches diversifiées de la question.

C'est pourquoi nous ferons cette hypothèse :

Les logiciels à communication II, tout spécialement du sous-type 3.2 II-I du type B, ou du sous-type 2 II- du type C, peuvent accompagner l'élève ou l'enseignant en illustrant des aspects de questions mathématiques.

Cette hypothèse se justifie :

– La communication de type II assure au moins la possibilité de production à la demande : l'utilisateur est en mode commande.

– D'autre part, les sous-types II-I et II- de la communication II présents en types B et C sont plus adaptés à une production rapide et

diversifiée sur un domaine restreint (celui de l'illustration), que le sous-type II+ : nous avons déjà signalé que leurs instructions ont un moins grand caractère de généralité que celles du sous-type II+ ; elles correspondent à des paramétrages de l'objet ou à des interventions par sous-objets et opérations et n'en imposent pas une description importante comme en sous-type II+ ; enfin, la tâche de communication incombant à l'utilisateur est bien plus réduite qu'en sous-type II+ (en particulier, lexique minimum et surtout syntaxe généralement inexistante, les instructions de parcours en tenant lieu).

Toutefois, le sous-type 2II+ du type C ou le sous-type 3.1 II-I du type B peuvent éventuellement répondre à la problématique d'illustration :

– Le sous-type 2II+ du type C permet d'illustrer, mais ne présente pas de garantie en matière de simplicité et de rapidité d'obtention. Remarquons qu'il peut aussi, en raison du mode différé, être employé pour la réalisation d'un logiciel paramétrable pour l'illustration ; il convient alors de ne pas confondre : celui-ci est généralement de type 2II-, quel que soit le logiciel support de son écriture.

– le sous-type 3.III-I peut être impliqué pour l'illustration de démarches portant sur un objet prédéfini par le logiciel : le sous-type 3.1 de CONT ne permet pas (sauf détournement par un usage circonstanciel du logiciel) comme les sous-types 3.2 et 2 de CONT une décision de l'utilisateur à propos de l'objet à afficher.

L'illustration géométrique est la plus représentée.

Nous avons déjà dit qu'il était effectivement plus aisé de réaliser un mode commande dans ce domaine.

Des équipes de recherche (IREM CREEM) ont principalement travaillé sur des logiciels d'illustration de type B qu'ils nomment « imagiciels ». Ainsi, comme le signalait déjà le tableau récapitulatif de la page 121, un imagiciel est un logiciel à mode commande de type B (sous-type 3.2 II-I).

Nous suggérons ici une extension des logiciels d'illustration au type C (sous-type 2II-).

La revue Rencontres Pédagogiques (INRP) propose un numéro intitulé "Imagiciels" (n° 1, 1983) où l'on peut trouver des exemples. Nous citerons les logiciels FRISE, MOSAÏQUE, JOLYGNES en type B, et les logiciels PICTOR et COLORPAINT en type C.

Nous résumerons en soulignant que les logiciels d'illustration géométrique pertinents offrent de nombreuses images selon les valeurs de paramètres variés, voire des images difficilement obtentibles autrement, et

ceci avec une précision et une rapidité dépendant du matériel utilisé.

Peu de tentatives ont eu lieu dans le domaine numérique.

Nous pensons que, outre des logiciels qui restent à imaginer, les tableurs et les grapheurs offrent des possibilités intéressantes.

La section 7 de ce chapitre sera l'occasion de discuter de façon générale des possibilités des logiciels d'illustration, considérées du point de vue de l'élève.

6c Logiciels pour manipuler des textes

Nous évoquerons les possibilités offertes par les logiciels de traitement de textes.

Ces logiciels se classent globalement en sous-type 2II- du type C.

A mode commande, ils permettent d'écrire, de manipuler des textes.

Suivant le niveau d'enseignement, ils ne seront vraiment pertinents pour notre propos que lorsqu'ils disposent de jeux de caractères scientifiques.

Nous ne mentionnerons que pour mémoire l'intérêt qu'ils peuvent avoir pour l'enseignant dans la préparation de documents pour la classe : ceux-ci deviennent aisément actualisables, et sont, de par leur présentation, socialement intégrés.

Pour l'élève, plusieurs utilisations, que nous évoquerons sommairement, sont envisageables :

– Utilisation stricte

On connaît la difficulté qu'a l'élève à s'exprimer en mathématique, et l'importance de l'expression écrite : un traitement de textes peut aider l'élève, et aussi socialiser l'acte d'écrit.

L'enfant peut rédiger une solution, la reprendre : il dispose d'un brouillon toujours propre. Son texte s'apparente extérieurement à celui de l'enseignant, à celui d'un texte produit dans la société ; en outre, il est reproductible à de nombreux exemplaires.

– Utilisation élargie

Nous pensons que la réalisation d'objectifs méthodologiques relatifs à l'apprentissage de la résolution de problèmes pourrait être assurée en partie par des activités avec un traitement de textes : on pourrait imaginer que l'élève lise, réécrive, extraie les données pertinentes, les réorganise, dans un texte de problème écrit par l'enseignant ou d'autres élèves.

6d Logiciels pour stocker et gérer des informations

⇨ LOGICIELS DE GESTION

Ces logiciels se situent en type C (éventuellement B) ; ils proposent une communication de type II-, ou II+ si un générateur de programmes leur est associé.

Ils peuvent servir à l'enseignant ou à l'élève pour la création et/ou l'exploitation d'une banque de données.

Pour l'enseignant, il peut s'agir de créer et d'exploiter une banque documentaire modeste à usage interne (textes de problèmes pour la classe, ...), ou d'exploiter une banque plus ambitieuse, le cas échéant par réseau télématique.

Divers organismes dont l'APMEP, les IREM commencent à proposer des banques documentaires (activités, documentation, ...) accessibles par réseau télématique. EVAMA que nous avons déjà cité s'appuie sur EXOMA, banque documentaire qui, au 28-01-85 comptait comme premier module 212 exercices sur la proportionnalité (Rapport d'activités 84-85 - GRECO Didactique - Thème Informatique).

Pour l'élève, deux modes d'utilisation peuvent être envisagés :

- Utilisation stricte

L'élève consulte ou crée une banque de données.

Remarquons que la recherche de critères d'organisation ou de tri, peut ne pas être sans incidence sur le renforcement des savoirs, leur réorganisation. Il n'est pas non plus impossible que l'élève établisse des relations entre plusieurs de ses savoirs lors de l'interrogation d'une banque. L'institutionnalisation de notions peut y gagner.

- Utilisation élargie

Il s'agirait, généralement selon une consigne, d'intervenir sur une banque de données en utilisant les critères de tri comme langage déclaratif restreint. Ceci permettrait, dans des situations qui s'y prêtent, une approche des relations qu'entretiennent les objets de la banque consultée.

On peut aussi imaginer une base de données consultable par l'élève, composée de types de problèmes et d'algorithmes de solutions paramétrables : la fonction didactique peut être le choix du bon programme de solution dans le fichier "algorithmes de solutions".

⇨ LOGICIELS PERMETTANT L'ENREGISTREMENT D'ACTES DE L'ELEVE

Il existe des logiciels qui comportent une prise sur fichiers d'actes de l'élève pendant leur utilisation.

Ces fichiers présentent un intérêt didactique certain.

A notre connaissance, actuellement, leur constitution s'effectue à l'adresse de l'enseignant ; nous avons déjà signalé le problème que pose l'analyse de tels fichiers (voir p. 222).

Nous pensons que les renseignements qu'ils contiennent pourraient, le cas échéant, servir à la construction de situations que l'on proposerait à l'élève, comme feed-back de son activité.

6e En guise de conclusion

Les logiciels évoqués pour la spécificité d'accompagnement de l'enseignement ou de l'apprentissage sont de type B ou C, c'est-à-dire à mode commande.

Certains d'entre-eux sont spécialement conçus pour l'enseignement.

D'autres sont des logiciels professionnels ou semi-professionnels, dont l'enseignant ou l'élève fait un usage strict ou élargi ; une socialisation de l'apprentissage mathématique peut être due à l'emploi de tels logiciels.

Nous ne parlerons pas de prospective particulière de construction : en effet, la capacité d'illustration de notions des logiciels conçus à fin d'enseignement repose sur une spécificité que nous allons maintenant traiter.

7- PEUT-ON DEFINIR UN NOUVEAU CADRE - LE CADRE LOGICIEL - POUR DES CONCEPTS MATHÉMATIQUES ?

7a Position du problème

Nous employons le terme «cadre» au sens de R. Douady.

Un cadre est un lieu spécifique de traduction de concepts, et partant, de leur mobilisation.

On connaît par exemple, les cadres géométrique, numérique, graphique ou physique.

Nous allons rechercher si l'objet logiciel peut constituer d'une façon générale un nouveau cadre, ceci à quelles conditions et dans quelles limites.

Il ne s'agit donc pas d'une étude portant sur la capacité d'un logiciel à proposer des activités relatives à un concept dans un ou plusieurs des cadres connus.

Remarquons qu'il existe effectivement des logiciels faisant intervenir plusieurs cadres pour une notion : par exemple, le logiciel RECT, situe la désignation multiplicative dans un jeu de cadres numérique/graphique ; cette caractéristique possible des logiciels n'est pas celle que nous allons examiner.

7b Cadre logiciel et communication

Nous ferons une hypothèse :

Il est possible que la communication de type II constitue un nouveau cadre que nous appellerons cadre langage.

Toutefois, dans certains cas ou dans certains usages, la communication de type II n'établit qu'un cadre ou plutôt un système descriptif fonctionnant à l'intérieur des cadres usuels (géométrique, graphique, numérique, ...).

Nous allons discuter cette hypothèse, sans prétendre résoudre complètement les problèmes qu'elle pose.

L'argumentation que nous développerons tout au long de cette section est, en quelque sorte, une forme de relecture de l'ensemble de notre travail.

Rappelons que notre propos concerne essentiellement la communication à sémantique procédurale, mais s'étend vraisemblablement à celle à sémantique déclarative.

⇒ LA COMMUNICATION DE TYPE II EST SUSCEPTIBLE DE CONSTITUER UN CADRE

Tout d'abord, l'ensemble de notre recherche tend à montrer que si le domaine logiciel peut avoir statut de cadre, ce ne peut être qu'au niveau de la communication : conformément à notre hypothèse initiale, elle est un élément spécifique.

Ceci étant, un statut de cadre ne peut lui être reconnue que si elle est un lieu original de traduction et de "manipulation" de concepts.

La communication de type I ne peut prétendre à cela : elle n'offre pas de moyens internes d'expression, et en conséquence, aucune expression, ni même description, ne peut lui être attachée ; l'utilisateur ne dispose que d'un mode réponse dont le lexique est extérieur au logiciel.

Par contre, la communication de type II est régie par des règles opposées : les instructions de production, et le cas échéant la syntaxe de la communication, sont des moyens logiciels d'expression ; l'utilisateur est en mode commande.

Nous avons d'ailleurs signalé que l'activité de communication pouvait devenir prépondérante, et qu'un logiciel à communication II pouvait être un outil de formulation par excellence (cf p. 244-247).

Nous distinguerons maintenant un fonctionnement de la communication II en tant que cadre langage strict, d'un fonctionnement comme outil descriptif dans des cadres existants.

⇒ COMMUNICATION DE TYPE II ET CADRE LANGAGE

Nous allons essayer d'expliquer par un exemple ce que nous entendons en disant qu'un logiciel à communication II peut constituer un cadre langage.

Pour cela, nous ébaucherons une réflexion à propos du logiciel LOGO (sous-type II+) comme cadre langage pour certaines notions géométriques. Plus précisément, nous expliciterons notre propos en considérant ce que pourraient être des éléments d'un jeu de cadres langage/géométrique.

Il s'agit de voir en quoi les instructions et la syntaxe du langage LOGO peuvent réaliser un ou des systèmes de signifiés autonomes admettant des correspondants dans le cadre géométrique.

Dans ces conditions, l'étude indépendante d'un système du cadre langage pourrait conduire l'élève à des résultats dont la traduction dans le cadre géométrique fournit des énoncés non immédiats.

Voici deux énoncés relatifs à la notion d'isométrie du plan dans le cadre géométrique :

(E1) Il existe une isométrie positive du plan qui transforme la figure F en la figure G.

(E2) La figure F admet un centre de rotation d'ordre n.

Ils ont des correspondants dans le cadre langage, représenté par LOGO :

(E1) Il existe un même texte LOGO qui représente les figures F et G.

(E2) Il existe un texte LOGO représentant la figure F qui est de la forme :

REPETE n [M TD p],

où M est un texte LOGO tel que la somme algébrique des arguments des instructions TD et TG est congrue à zéro, modulo 360, et où np est congru à zéro, modulo 360.

Nous dirons que dans le cadre logiciel, (E1) exprime les éléments du groupe des isométries positives, tandis que (E2) caractérise pour une figure l'invariance par rotation d'ordre n.

Un texte étant donné, s'il est linéaire sans ordre REPETE, il est possible de tenter de le transformer en se conformant aux règles de la communication LOGO pour apprécier s'il admet une forme (E2), ou au contraire, s'il est de forme (E2) de le développer.

Ainsi, il est possible par manipulation de propriétés propres au cadre langage de construire des résultats transmissibles dans le cadre géométrique, ou réciproquement.

Soulignons que la problématique que nous venons d'exposer pourrait être ouverte à d'autres logiciels que LOGO, à sémantique procédurale, mais aussi déclarative : en particulier, on pourrait s'intéresser au langage PROLOG.

⇒ COMMUNICATION DE TYPE II ET CADRE DESCRIPTIF

Ici, il ne s'agit plus de traduction originale de concepts dans le domaine du langage, mais d'un usage de la communication de type II comme outil de description fonctionnant dans les cadres existants.

Un logiciel à communication II peut voir son rôle restreint à celui de cadre descriptif pour deux raisons :

– le sous-type de la communication II en cause n'offre pas de possibilité effective de cadre langage.

– l'élève n'est pas en mesure de travailler dans le cadre langage mis à sa disposition : il n'utilise, en appui sur les instructions de production, que les possibilités descriptives du cadre, qui se trouve ainsi réduit à un cadre descriptif.

Ce dernier point suggère une remarque : le cadre langage contient vraisemblablement le cadre descriptif.

Nous traiterons des conditions d'existence de ces deux contextes au cours des paragraphes suivants.

⇒ SOUS-TYPES DE LA COMMUNICATION II, CADRE LANGAGE, CADRE DESCRIPTIF

La communication de type II présente trois sous-types : les sous-types II-I, II-, II+.

Nous les avons déjà distingués pour certains critères :

– En sous-type II-I, les instructions n'ont pas un grand caractère de généralité, sont souvent relatives à des objets déterminés (objets, sous-objets, paramètres d'objets). Et surtout, des instructions de parcours règlent le discours de l'utilisateur.

Pour ces raisons, les possibilités d'expression et d'écriture de textes sont donc limitées.

Par conséquent, nous dirons que le sous-type II-I n'offre pas de garanties suffisantes pour procurer un cadre langage ; par contre, il pourra fonctionner comme cadre descriptif.

– En sous-type II+, de larges possibilités existent : les instructions sont généralement indépendantes d'objets particuliers (sous-objets "neutres" et opérations), les représentations de données peuvent être multiples, la syntaxe relativement riche. Il n'y a pas d'instructions de parcours.

La complexité même de l'organisation de la communication dans ce sous-type, engage à penser que de véritables textes peuvent être produits, et transformés de manière interne, sans sortir du domaine de la communication.

Par conséquent, nous pensons que le sous-type II+ peut constituer un cadre langage.

Le mode différé ne nous paraît pas en être le fondement ; toutefois, il peut offrir des possibilités accrues, et surtout provoquer spécialement l'élève à mener des activités temporairement à l'intérieur du cadre langage : écriture et transformation de textes, ce qui pourra le conduire ensuite à des résultats dans un cadre ou un autre.

– Le sous-type II- se présente de manière analogue au sous-type II+, mais dans une moindre mesure.

Les instructions de production demandent moins de description, mais restent distinctes d'objets spécifiques ; il n'y a pas d'instructions de parcours.

On pourra consulter le descriptif de la typologie COM (p.98-103) et le chapitre "Logiciels et communication : compléments" (p. 124-132)

C'est pourquoi, nous pensons qu'il est possible que le sous-type II- fonctionne comme cadre langage.

Toutefois, l'absence de mode différé peut être un handicap, d'autant que l'interface souvent graphique (pictogrammes) peut distraire l'utilisateur d'un travail d'étude de l'écrit qu'il a conçu. En outre, le caractère souvent rudimentaire de la syntaxe peut éloigner du cadre langage, dans le même temps où il permet une utilisation plus aisée (moins d'apprentissage de la communication) et plus directement en relation avec un objet mathématique.

7c Cadre logiciel et concepts mathématiques

Nous ferons une rapide synthèse des résultats du chapitre "Transposition de concepts mathématiques et logiciels" (p. 198-209) et du précédent paragraphe "Sous-types de la communication II, cadre langage, cadre descriptif".

Nous avons pris, pour expliciter le cadre langage, un exemple d'ordre géométrique quant au contenu mathématique : ce n'est pas un hasard ; en effet, rappelons qu'il est plus aisé de définir des instructions concernant des questions géométriques.

L'étude faite au chapitre "Transposition de concepts mathématiques et logiciels" (p. 198-209) va dans ce sens : pour le niveau considéré (l'école élémentaire), et pour des logiciels à sémantique procédurale, il est plus aisé de définir un mode commande concernant des questions géométriques ; toutefois, rappelons que nous avons également émis en page 201 une hypothèse relative à la pertinence des logiciels à sémantique déclarative pour l'étude des relations qu'entretiennent entre eux les objets du domaine numérique.

Plus précisément :

Actuellement, il est vraisemblable de trouver pour l'expression du domaine géométrique, des logiciels :

- cadre langage (et a-fortiori, cadre descriptif), en sous-type II+ ou II-
- cadre descriptif (voire, cadre langage faible), en sous-type II-I.

En outre, il n'est pas impossible de trouver pour l'expression du domaine numérique des logiciels, cadre descriptif (voire, cadre langage faible), en sous-type II-I.

Une étude serait à mener avec les logiciels existants concernant l'intervention dans un cadre langage de concepts mathématiques : quels sont les concepts concernés ? Pour lesquels de leurs aspects ? Comment la traduction s'effectue-t-elle dans le cadre langage ?

Par ailleurs, une problématique de construction de logiciels, à sémantique procédurale ou déclarative, est ouverte.

7d Les élèves de l'école élémentaire et le cadre langage

Les élèves de l'école élémentaire peuvent-ils utiliser un cadre langage pour un apprentissage mathématique ?

Nous relaterons tout d'abord une observation faite auprès d'instituteurs en formation :

Considérons en LOGO le texte REPETE 4 [AV 50 TD 90].

Dans le cadre langage, on conclut directement que ce texte traduit une figure admettant un centre de rotation d'ordre 4 (cf les énoncés (E2) et (E2')) ; l'élément minimum, générateur de la figure par rotation, correspond à AV 50 : il s'agit d'un segment qui n'est pas un circuit fermé ; finalement, la figure exprimée par le cadre langage est un carré plutôt qu'une croix (+). Rien ne renseigne a priori sur la position du centre de rotation, mais une étude supplémentaire pourrait le faire, notamment dans des cas particuliers.

Apprécier d'emblée ce texte comme carré potentiel suppose une connaissance du carré en terme de figure globalement invariante dans une rotation d'ordre 4.

Lorsque l'on demande à des instituteurs en formation d'expliquer, de valider leur affirmation relative à l'expression d'un carré par un tel texte, ils invoquent TD 90 et le nombre 4, AV 50 et le nombre 4, pour affirmer l'existence respectivement de quatre angles droits, et de quatre côtés de même longueur.

Selon nous, ceci signifie qu'ils n'appréhendent que de façon partielle le cadre langage, au sens où ils n'y perçoivent pas directement la traduction d'objets mathématiques.

En fait, ils tentent de reconstituer des propriétés connues du cadre géométrique ; ils développent mentalement le texte pour en obtenir une expression équivalente correspondante à la procédure de description de l'objet, qu'ils ont eux-mêmes suivie (malheureusement, nous n'avons pas eu l'occasion de soumettre des textes différents de productions effectuées, ou éloignées de celles-ci ; une expérimentation serait nécessaire pour prolonger les observations citées).

Remarquons que l'on assiste néanmoins à un fonctionnement interne au cadre langage (développement du texte), que l'on

pourrait presque qualifié de régressif. Un énoncé de type (E'2) n'admet pas pour ces instituteurs de signification géométrique ; le terme REPETE ne constitue qu'une abréviation, dont l'emploi est souvent automatisé. Ainsi, si on leur propose d'écrire un texte pour une rosace d'ordre 5, ils ne produisent pas directement REPETE 5 [m AV 360/5], ou alors au niveau d'un réflexe qui ne s'opère plus pour des objets complexes.

Ceci peut expliquer les problèmes rencontrés à propos de la réalisation de certaines figures : l'utilisateur ne procède que par simple "reconnaissance de formes" ; il reste au niveau d'une description non organisée par un savoir, en particulier, par un savoir relatif à la traduction de l'objet mathématique dans le cadre langage ; ce faisant, si le problème n'est pas élémentaire, sa résolution n'est pas assurée.

L'hypothèse proposée ci-dessus n'est pas sans rappeler des résultats obtenus par A. Robert lors de son étude de l'acquisition du concept de convergence des suites numériques :

Les étudiants expriment sur la convergence des suites numériques trois grands types de représentations : "dynamiques", "statiques", "monotones". Ceux d'entre eux qui recourent à une représentation statique, c'est-à-dire à "une formulation en langue naturelle de la définition formalisée de la convergence d'une suite" (c'est-à-dire proche de celle-ci), associée ou non à une formulation "dynamique" (où "converger" signifie "se rapprocher de"), réalisent les meilleures performances.

Ainsi, dans le domaine de la convergence des suites, comme dans celui de la production de figures géométriques en LOGO, une représentation organisée par un savoir, mettant en jeu une traduction de l'objet dans un langage ad hoc, paraît garante d'une plus grande réussite qu'une représentation imagée, simple reconnaissance de formes.

(On pourra consulter :

- "Acquisition de la notion de convergence des suites numériques dans l'enseignement des mathématiques", A. Robert, Thèse d'Etat, Université Paris VII, 1982.
- "Acquisition de la notion de convergence", A. Robert, Revue Recherches en Didactique des Mathématiques Vol 3.3, 1982.
- "Ingénierie didactique sur les suites numériques après le baccalauréat", A. Robert, Cahier de Didactique des Mathématiques n° 4, Publication IREM Paris VII.)

Les lignes précédentes nous suggèrent des réflexions :

Utiliser pour un objet mathématique le cadre langage, c'est repérer une formulation comme significative de cet objet, ou produire directement une formulation de cet objet.

Ceci suppose, en particulier, un autre point de vue sur l'objet qu'en terme de séquentialité d'actions.

Par exemple, pour LOGO, utiliser le cadre langage, n'implique-t-il pas de pouvoir se représenter le texte comme un tout, de le lire en terme statique comme signifiant d'un objet géométrique ?

On est alors conduit à considérer des relations entre les cadres concernés, plus précisément entre éléments du texte et propriétés de l'objet exprimées par le texte, de façon statique.

Reprenons le texte REPETE 4 [AV 50 TD 90] :

Ayant vérifié que la forme de ce texte est celle de l'énoncé E'2 (voir p. 256), c'est-à-dire, en particulier, que la somme algébrique des arguments de TD et TG dans l'élément de texte AV 50 est congrue à zéro, on identifie une figure globalement invariante dans une rotation d'ordre 4.

En outre, le texte précise un système générateur minimum de la figure :

AV 50 correspond à l'expression d'un élément minimum permettant d'engendrer la figure.

TD 90 correspond à l'expression de la nature d'une transformation génératrice.

REPETE 4 correspond à l'expression de l'ordre, ou encore à l'expression de l'invariance.

Intervient ainsi une connaissance d'ordre mathématique, une connaissance de la communication, une mise en relation de deux cadres.

Par conséquent, nous pensons que ceci impose des enseignements spécifiques. Or, à notre connaissance, l'enseignement relatif à la traduction de concepts dans un cadre langage, c'est-à-dire à la mise en relation de cadres, fait particulièrement défaut. Il pourrait peut-être exister, ne serait que pour LOGO et les concepts géométriques.

Actuellement, les élèves de l'école élémentaire, même de cycle moyen ne peuvent, selon nous, utiliser que très partiellement un véritable cadre langage.

Il y a un problème d'accessibilité au cadre même : nous avons traité de la complexité des tâches de communication (surtout en sous-type II+), et la délicate dialectique apprentissage mathématique/apprentissage de la communication.

Nous renvoyons à l'étude que nous avons faite des tâches de communication II (p. 139-154).

Rappelons qu'il existe des recherches concernant l'accessibilité de LOGO pour des enfants de l'école élémentaire (P. Mendelsohn), et d'une façon plus générale des recherches en didactique de l'informatique (N. Balacheff, J. Rogalski, A. Rouchier, R. Samurçay) ; enfin, le problème posé par l'enseignement de l'informatique est également reconnu par l'existence à

l'Université Paris VII d'un enseignement de didactique de l'informatique assuré par J. Arzac, en 3^e cycle de Didactique des mathématiques et de l'informatique.

En outre, les acquis mathématiques de l'élève ne sont peut-être pas suffisants pour aborder des concepts dans le cadre langage : en effet, il nous semble improbable que l'enfant puisse considérer les objets mathématiques de façon formelle, en établissant une unification d'ordre structurel, ce qui paraît nécessaire à l'usage du cadre langage, si l'on se réfère aux hypothèses que nous avons faites à propos de LOGO.

Nous avons souligné plus haut la nécessité d'une appréciation globale, statique, du texte et de l'objet géométrique qui lui correspond ; une lecture non linéaire de l'objet, mais structurée, par des propriétés caractéristiques doit être disponible ; on pourrait dire en quelque sorte, qu'une connaissance mathématique non élémentaire de l'objet est requise.

C'est pourquoi, une expérimentation relative à un enseignement plaçant un concept dans le cadre langage nous paraît nécessaire.

Un tel enseignement pourrait permettre, si ce n'est une prise en compte complète par l'élève du cadre langage, un éloignement mesuré d'un emploi de la communication de type II de genre essentiellement descriptif : à l'heure actuelle, l'élève utilise le cadre descriptif inclus dans le cadre langage.

Les élèves pourraient constater des invariants dans des textes, exprimer des remarques, travailler à l'élaboration restreinte de classes d'objets mathématiques et de relations entre celles-ci.

D'autre part, l'enseignant pourrait être attentif à un usage de la communication en conformité avec le cadre langage ; il serait intéressant d'apprécier la capacité d'un texte à traduire de façon organisée une notion : par exemple, en LOGO, une programmation avec REPETE pour une rosace, et une formulation de type récursive pour une frise respectent de façon satisfaisante les notions mathématiques.

En outre, on peut penser à "globaliser" les éléments du langage pour faciliter la traduction cadre géométrique/cadre langage.

Le lecteur pourra consulter l'article "Activation de schémas de programmation et mémorisation de figures géométriques" de P. Mendelsohn (Journal européen de psychologie de l'éducation Vol 1 n°2, 1986).

Il n'est pas impossible que se construisent ainsi, par une approche descriptive, des résultats dans un jeu de cadres, cadre langage/cadre X.

7e Les élèves de l'école élémentaire et le cadre descriptif

Cet aspect du cadre logiciel offert par la communication de type II est, dans l'état actuel des choses, le plus à la portée des enfants de l'école élémentaire.

L'ensemble de notre recherche tend à montrer qu'il est possible qu'un logiciel-cadre descriptif permette des acquisitions dans une dialectique outil-objet, à condition qu'une certaine maîtrise de la communication soit assurée, ou qu'une dialectique adaptée apprentissage de la communication/apprentissage mathématique intervienne.

Nous ferons tout d'abord des remarques générales.

Il n'est pas impossible que, de la même façon que le cadre descriptif peut faire écran au cadre langage, le langage naturel ou les connaissances construites dans d'autres cadres constituent des obstacles au cadre descriptif.

Ainsi, un enfant a tenté de calquer en LOGO sa méthode de construction sur papier d'un rectangle (dessin des deux largeurs, puis des deux longueurs). L'échec a été incompris dans un premier temps.

Signalons aussi que, si l'on souhaite que l'élève utilise un logiciel comme outil de description, il est préférable que les instructions de production apparaissent en tant que telles, et non pas comme options parmi lesquelles il faut effectuer un tri :

- le type C répond bien à cette exigence avec des logiciels langages de programmation ou de commande, accompagnés d'une documentation explicite concernant la communication.

- par contre, le type B qui propose également des langages de commande, plus restreints, ne satisfait que rarement à ce vœu : les instructions licites sont souvent présentes à l'écran, et peuvent être comprises comme des options ; l'utilisateur effectue ses entrées généralement suite à des instructions de parcours-répérage ; la qualité "instructions de production" des entrées permises risque de passer inaperçue et l'utilisateur peut se considérer en mode réponse.

Peut-être conviendrait-il d'y remédier dans la mesure où, un tel vécu du logiciel par l'élève en type I (mode réponse), peut s'avérer peu propice à des initiatives de description ; le logiciel ne serait donc pas pleinement employé.

Nous allons maintenant compléter les résultats des chapitres précédents en précisant ce qu'il est possible d'attendre de l'emploi d'un logiciel-cadre descriptif.

En quelque sorte, nous allons tenter d'explicitier un support de

l'éventuelle acquisition dans une dialectique outil-objet.

⇒ INTELLECTUALISATION DE L'ACTION

Avec un logiciel-cadre descriptif, l'action de l'enfant, d'un point de vue extérieur, consiste à donner une ou plusieurs instructions. Pour obtenir la production désirée, même en l'absence de mode différé, l'élève est ainsi contraint à une intellectualisation préalable de son action ; il doit anticiper, même s'il ne le fait que localement.

Prenons quelques exemples pour éclairer notre propos :

- Nous citerons une analyse relative au logiciel SIMUL (type B), à laquelle nous avons participé :
"- avec une grille de jetons : l'élève peut manipuler directement les jetons mais son action porte sur des jetons séparés et non sur la grille complète ;
- avec une grille dessinée sur une feuille de papier : l'élève ne peut réaliser la manipulation précédente, il ne peut pas (ou difficilement) réorganiser la collection et de ce fait, son exploration est limitée (car essentiellement visuelle).
- avec une grille sur ordinateur : l'élève est mis en demeure d'anticiper l'action qu'il aurait faite pour recouvrir une grille dessinée sur du papier, avec des jetons. Il doit l'explicitier car il doit commander l'affichage de cette grille à la machine."
("Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire." Thèse de 3^e cycle D. Butten).
- Le logiciel PUZZLE (type B) se distingue des puzzles en carton pour des raisons analogues : la manipulation avec les mains des pièces d'un puzzle en carton autorise l'enfant à opérer des placements et des corrections locales (retournement, glissement de pièces) sans qu'il soit forcé à une stricte anticipation l'amenant à une description de la résolution du problème. Avec le logiciel, choix et placement d'une pièce au crayon optique correspondent à un pointage de l'action au niveau conceptuel.
- Avec le logiciel DIVIS (type B), il en est de même : l'élève tente rapidement de prévoir le résultat de l'opération qu'il désignera au crayon optique.

Rappelons que suite à la production, l'élève peut réfléchir à son action selon ce qu'il constate, effectuer des validations locales dans des limites déjà étudiées. Nous partageons à ce propos une analyse de A. Rouchier selon laquelle il s'agit moins d'une réflexion de l'enfant sur sa pensée que, éventuellement, sur son action : la théorie de l'enfant épistémologue est à nuancer.

⇒ LIMITES DE L'INTELLECTUALISATION DE L'ACTION

Signalons tout d'abord, qu'au niveau de leurs instructions, les logiciels des types B et C offrent, dans le domaine de l'anticipation, les mêmes potentialités.

Il est toutefois vrai que l'existence d'un mode différé en type C peut mieux contraindre l'élève, à condition que l'enseignant en impose le cas échéant l'emploi : le recours au mode différé qui suppose une capacité importante d'anticipation, ne s'opère pas naturellement, comme nous allons le voir.

Les instructions de parcours d'un logiciel de type B peuvent constituer une gêne pour l'anticipation ; toutefois, on peut envisager de demander à l'élève de préparer le texte à transmettre, ou pourquoi pas, ultérieurement une transformation du logiciel.

Comment l'élève dispose-t-il des possibilités offertes ?

Nous avons observé que l'élève n'anticipe pas spontanément.

Une expérience en maternelle a montré que de jeunes élèves échouent à diriger une tortue de soi LOGO sous une maison, sans la voir : ils ne peuvent anticiper correctement.

Avec des élèves plus âgés, en LOGO, on constate que l'écriture et l'emploi de procédures ne sont pas adoptés sans réticence : la production par appel d'une procédure est pour eux peu convaincante, même s'ils ont rédigé le texte de la procédure.

Nous proposerons une explication à cela : l'affichage automatique à partir d'un mot (le nom de la procédure) a pour eux un côté magique ; il n'est pas impossible que cela aille contre un mode de pensée dynamique, plus proche de descriptions de genre linéaire.

Une des difficultés de l'apprentissage d'un langage autorisant l'intervention de procédures est ici soulignée ; l'utilisateur doit être capable de deux points de vue sur un même objet : un point de vue descriptif dynamique au moment de la création de la procédure (même s'il opère de façon structurée), et un point de vue global et statique au moment de l'appel de la procédure.

Notons en outre, que la tâche de l'élève en mode différé est accrue : l'identification de l'occurrence dans un texte d'une instruction inadaptée est souvent difficile ; en mode direct, la recherche de l'endroit défaillant peut être évitée.

Ainsi, les élèves de l'école élémentaire sont limités en ce qui concerne l'anticipation et la description.

Ils n'utilisent que peu les offres d'un mode différé puissant, car celles-ci ne sont pas à leur portée : la communication implique des actes trop complexes, mais aussi, les objets à décrire sont souvent insuffisamment connus du point de vue de leur contenu.

De ce fait, avec un logiciel-cadre descriptif, c'est la linéarisation d'un processus à propos d'une situation singulière avec valeurs numériques (ni paramètres, ni tests) qui intervient, plutôt que l'expression d'un algorithme de résolution d'une véritable classe de problèmes.

Selon nous, cela reste néanmoins performant, car difficilement envisageable avec la même rigueur dans la langue naturelle.

L'obligation de recours, soit au mode direct, soit au mode différé (quand celui-ci existe), doit être un objectif raisonné, compte-tenu des difficultés soulignées.

⇒ A PROPOS DE L'OBJET MATHÉMATIQUE D'ENSEIGNEMENT

Le cadre descriptif permet à l'élève d'exprimer dans une certaine mesure, sa représentation du problème ; par là-même, et dans le cadre des instructions, il désigne les paramètres qui lui paraissent pertinents.

Nous avons d'ailleurs déjà signalé que des logiciels pouvaient être produits dans le but de forcer des procédures ou des lectures particulières.

Il n'est pas nécessaire que le logiciel soit un interpréteur au sens strict : rappelons que des logiciels de type B, régis par une syntaxe sommaire, voire inexistante, mais autorisant des paramétrages sont peut-être plus propices à engendrer un travail de mathématique expérimentale.

En fait, les lectures d'objets permises par un logiciel sont toujours orientées par les instructions disponibles, qu'il s'agisse d'un logiciel de type B ou de type C : par exemple, en LOGO (type C), une figure est un parcours orienté.

Ainsi, les systèmes d'instructions d'un logiciel cadre-descriptif placent les objets d'enseignement dans certains cadres mathématiques, et n'y permettent que certaines procédures.

Par conséquent, le cadre descriptif peut constituer une aide, s'il permet de focaliser sur des éléments pertinents de l'objet d'enseignement, ou au contraire un obstacle s'il détourne de ceux-ci, voire accentue des

représentations non propices ou erronées de l'élève.

Une expérimentation ponctuelle a montré que les instructions de repérage local en LOGO, sont pertinentes pour forcer à la mise en évidence du centre de rotation d'une rosace non connexe : elles font échouer une production basée sur la lecture spontanée que fait l'élève de la rosace par positions relatives de ses branches. Par contre, lors de la reproduction d'une frise, l'élève est confirmé dans sa lecture par positions relatives et écart des motifs : dans ce cas, LOGO accentue l'incapacité d'une lecture de type "transformation par translation".

En conclusion, nous dirons que le cadre descriptif attaché aux logiciels B et C permet la définition de situations-problèmes que l'élève traite avec ses savoirs et les éléments de la situation, c'est-à-dire les instructions, et les divers autres moyens logiciels.

En particulier, les micromondes dont il est question dans certains ouvrages relatifs à LOGO sont, dans le meilleur des cas, des situations-problèmes ouvertes.

L'objectif de la situation-problème peut consister en la construction d'une connaissance nouvelle, sujette à institutionnalisation, ou encore en une épreuve de connaissances, plus large que de la simple répétition.

Une situation-problème mise en place par un logiciel à communication II, comme toute situation-problème, si riche soit-elle, privilégie un ou quelques aspects d'un concept.

A partir d'une description, on peut formuler certaines propriétés, mais pas toutes.

Bâtir des définitions, placer le concept dans un réel savoir supposent des activités variées, dont certaines se dérouleront nécessairement sans machine.

Considérons ce texte LOGO :

REPETE 2 [AV 50 TD 90 AV 30 TD 90]

On peut se ramener à un énoncé de type E2 :

REPETE 2 [AV 50 TD 90 AV 30 TD 90 TG 180 TD 180]

En supposant même que l'élève y reconnaisse un rectangle, et pour celui-ci l'existence d'un centre de symétrie en raison de la formulation, il n'est pas évident que soit automatiquement associée la propriété des diagonales de se couper en leur milieu.

D'autres cadres que le cadre logiciel sont nécessaires, d'autant que le cadre langage peut figer par une dénomination un concept sur la description qu'il en permet (par exemple, un carré devient le parcours orienté écrit pour la procédure CARRE).

Ceci indique qu'une bonne gestion didactique suppose l'existence de l'enseignant : le cadre descriptif favorise l'acquisition de savoirs par l'action dans une dialectique outil-objet ; l'enseignant intervient pour choisir une situation adaptée, pour susciter formulation et validation, pour évaluer, pour institutionnaliser, ... et finalement aider à dégager le concept de sa description dans le cadre logiciel.

7I En guise de conclusion

Il est possible de définir une prospective de construction de logiciels cadre-langage, ou surtout cadre-descriptif.

De tels logiciels nous paraissent riches de possibilités didactiques.

La communication proposée doit être de type II.

Il n'est pas nécessaire qu'un mode différé existe ; toutefois, il peut utilement forcer à l'anticipation.

Un fonctionnement d'ensemble (instructions, syntaxe, représentation des données), relativement élémentaire du point de vue de la stricte communication est souhaitable, si l'on veut centrer la réflexion de l'utilisateur sur des objets disciplinaires. Il est aussi préférable que des instructions de parcours ne jalonnent pas son discours.

Dans ces conditions, nous pensons à nouveau que des mini-interpréteurs spécialisés, logiciels proches du type B et du type C (sous-type 2II-) seraient appréciables. Des énoncés de problèmes avec possibilités d'évaluation pourraient, le cas échéant, leur être adjoints.

Nous n'oublions pas l'apparition récente, hors secteur éducatif, de logiciels se classant dans ce sous-type 2II- (traitements de textes, tableurs, SGBD, ...). Leur succès est peut-être précisément dû à ce que nous demandons ici pour l'enseignement.

CONCLUSION



1- A PROPOS DU DOMAINE DE VALIDITE DE LA RECHERCHE

Nous nous sommes intéressée essentiellement aux logiciels à sémantique procédurale, mais notre propos semble susceptible de s'appliquer aux logiciels à sémantique déclarative.

Nous avons choisi de conduire notre étude au niveau "Ecole Élémentaire", et nous avons donc proposé des exemples pris à ce niveau.

Toutefois, l'analyse menée nous paraît montrer que le domaine de validité des résultats est plus large : les résultats se conçoivent pour d'autres niveaux d'enseignement, et pour nombre des questions abordées, moyennant le cas échéant quelques ajustements, à d'autres disciplines.

2- A PROPOS DE LA COMMUNICATION PROPOSEE PAR UN LOGICIEL

Notre recherche met en évidence que la communication avec le micro-ordinateur est un élément spécifique, que l'on peut considérer comme une variable didactique.

Il existe deux grands types de communication :

- Le type I de communication où l'utilisateur est en mode réponse ; une situation lui étant proposée, aucun moyen logiciel ne lui permet de la traiter.

- Le type II de communication où l'utilisateur est en mode commande ; une situation lui étant proposée, des moyens logiciels, les instructions de production employées selon une syntaxe appropriée, lui permettent de la traiter.

Certains logiciels se classent sans équivoque en type II de communication. D'autres présentent des caractéristiques (par exemple, émission d'une consigne d'activité) rendant leur affiliation plus délicate.

C'est pourquoi, nous énumérerons ici en synthèse, divers moyens d'identification du type II de communication pour un logiciel.

Avec un logiciel à communication II, c'est-à-dire à mode commande :

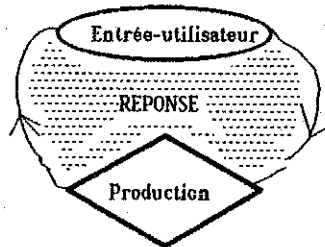
- On peut dresser et expliciter la liste des instructions de production et les règles de leur emploi, avant une utilisation effective du logiciel.

- Les instructions ont un sens indépendamment du problème posé.

- On peut, a priori, définir un autre problème que celui qui est posé et

y répondre (la question n'est ici celle de la pertinence, mais de la possibilité intrinsèque de définition différente).

- L'utilisateur peut intervenir sur la définition du problème.
- L'utilisateur peut enchaîner plusieurs entrées faites avec des instructions ; le logiciel peut toutefois émettre entre celles-ci une instruction d'invite à effectuer une entrée.
- Il n'y a pas à chaque entrée de l'utilisateur une évaluation par rapport à l'attendu conceptuel.
- L'organisation du logiciel s'apparente à :



- L'utilisateur peut exprimer avec les instructions de production sa propre représentation du problème.
- L'utilisateur peut recourir à un fonctionnement par essai-erreur.

Nous soulignerons que le développement actuel des logiciels, la façon dont il en est question dans le discours social nous semblent établir une forme de preuve de la validité du point de vue que nous avons adopté dès 1982 : ce point de vue d'analyse des logiciels selon la communication, — et plus précisément en référence aux deux grands modes cités —, qui constitue le fondement de notre recherche, s'actualise.

Ainsi, dans le domaine extra-scolaire, des logiciels que nous situons en communication II comme des tableurs, des grapheurs, ... sont maintenant en pleine expansion et reconnus comme des langages, presque au même titre que des interpréteurs. Cette position n'était pas acceptée au début de nos travaux. Mieux encore, les logiciels qui se réalisent paraissent présenter presque systématiquement une interface communication avec l'utilisateur de type II, c'est-à-dire à mode commande, et par conséquent avoir été projetés selon ce principe.

Qu'en est-il aujourd'hui dans le secteur éducatif?

La reconnaissance du type II de communication pour un logiciel, c'est-à-dire son analogie de fonctionnement avec un logiciel comme LOGO du

point de vue de la communication, s'accomplit diversement.

Les tableurs, traitements de textes, logiciels graphiques sont assez aisément apparentés aux langages.

Par contre, d'autres logiciels qui fonctionnent selon nous en type II de communication, mais avec des possibilités moindres, sont encore souvent affiliés à tort au type I de communication, d'après des caractéristiques extérieures, dont celle de définition partielle du problème à traiter.

Ceci nous paraît dommageable à un bon emploi didactique des moyens logiciels.

En effet, d'après les résultats de notre recherche, la communication a, au niveau didactique, un rôle primordial que nous rappellerons dans les lignes qui suivent : discerner le type de communication, et plus généralement le type du logiciel, peut aider à la mise au point de situations d'enseignement.

La communication a statut de variable didactique : on peut (ou on pourra ultérieurement) choisir pour une question donnée un logiciel d'un type déterminé de communication ; nous avons donné des indices spécifiant que le choix d'une communication particulière modifie les rapports des élèves, si ce n'est avec les notions en jeu (l'expérimentation est encore insuffisante), tout au moins avec la situation.

La communication se manifeste à un autre niveau en influant, dans une mesure qu'une prospective de construction de logiciels pourrait permettre plus tard de dépasser, sur l'organisation du contenu de l'activité proposée.

Ainsi, nous avons distingué au sein d'une typologie générale a priori, du double point de vue de la communication et du contenu, trois types de logiciels : le type A attaché au type I de communication, et les types B et C attachés au type II de communication.

En fait, le type B se différencie essentiellement du type C par définition partielle des activités proposées. Quelques logiciels de type A indiquent une tendance à la création possible d'un nouveau type ou sous-type, toujours à communication I, que nous avons désigné au cours de la rédaction par type A non-strict.

3- COMMUNICATION ET ENJEUX DIDACTIQUES DES TROIS TYPES DE LOGICIELS

La communication est également significative d'enjeux didactiques de l'utilisation effective pendant l'enseignement d'un logiciel.

Pour la clarté de l'exposé, et la mise en valeur des différences, nous résumerons ces enjeux successivement pour les types A, C, B.

3a Préalable concernant les trois types de logiciels

- La communication crée des tâches pour l'élève, distinctes suivant le type du logiciel, mais toujours effectives.

Ces nouvelles tâches ne devraient plus rester l'objet d'un contrat implicite.

- Tous les logiciels interviennent sur l'activité de l'élève, par la communication, spécifiquement selon celle-ci ; il ne s'agit pas uniquement d'un accroissement de la tâche par la tâche de communication.

Seuls les logiciels de type A, et quelquefois B contiennent des éléments d'intervention, autres que la communication.

3b Les logiciels de type A

⇒ LES LOGICIELS DE TYPE A STRICT

Ils sont réservés à un usage individuel. En effet, ils jouent auprès de l'élève les rôles ordinaires de l'enseignant et des autres élèves de la classe (donnée de consignes, interventions spécifiques, et surtout évaluation). Ils referment l'élève sur l'activité qu'ils définissent.

L'élève ne dispose que d'un mode réponse. Celui-ci ne lui permet pas d'argumenter.

Son rapport à la machine consiste à vouloir la vaincre en gagnant au «jeu» instauré.

Les logiciels de type A strict permettent au cours de situations qui ne sont pas des situations-problèmes :

- des apprentissages de savoirs nouveaux, mais limités, et ceci par un fonctionnement de type «mécanique objet-outil» ;
- des répétitions de savoirs anciens ;
- des évaluation de savoirs anciens.

La phase de validation de l'activité est inexistante : elle n'est prise en compte ni par l'élève, ni par le logiciel pour l'élève.

⇒ LES LOGICIELS DE TYPE A NON-STRICT

Ils constituent des ouvertures.

Dérageant à certains critères du type A (absence d'évaluation ou de corrigé de réponse, absence de formulation de consignes à l'écran, proposition de problèmes non-isomorphes, intervention d'éléments originaux de gestion comme la temporisation, présence d'une communication de type II pour la

réalisation d'un objet intermédiaire), ils offrent a contrario des libertés didactiques.

Ils peuvent proposer des situations-problèmes, qui traitables individuellement, sont toutefois susceptibles de s'insérer dans une activité d'ensemble.

Leur usage peut se justifier :

- à fin de préparer l'introduction et l'institutionnalisation de notions nouvelles par l'épreuve de l'inadaptation des savoirs existants ;
- à fin de mobilisation de savoirs ou de stratégies ;
- à fin de contrainte de l'élève, pertinente eu égard à l'objet d'enseignement.

⇒ EN CONCLUSION

En aucun cas, les logiciels de type A ne nous paraissent susceptibles de constituer un cadre (au sens de R. Douady), lieu spécifique de traduction de concepts mathématiques.

Compte-tenu de nos choix didactiques, ils nous intéressent pour des emplois conformes aux fonctions signalées. Ils peuvent, en ce sens, rendre plus efficace l'enseignement.

Les logiciels de type A strict n'interviennent pas pour une construction des savoirs par l'action. Leur usage peut ne se justifier que pour des questions restreintes, en particulier de type algorithmique, suite à des acquisitions préalables. Les logiciels de type A non-strict ouvrent un champ plus large, et peu connu.

Une prospective de construction de logiciels A strict pourrait consister à différencier sur des logiciels distincts les fonctions d'évaluation, de répétition, ou d'acquisition par mécanique «objet-outil».

Pour les logiciels A non-strict, une problématique de développement pourrait être, dans le but de réaliser une contrainte didactique pertinente sur l'activité de l'élève, la suppression de critères du type A, et l'adjonction d'éléments d'intervention.

3c Les logiciels de type C

Ils offrent des possibilités que l'on pourrait presque qualifier d'opposées à celles du type A.

La seule intervention du logiciel est celle due à l'effet de la communication de type II.

L'absence d'intervention explicite, notamment l'absence d'évaluation, et finalement la non-prise en charge directe de l'activité par le logiciel en font des outils didactiques.

Ainsi, ils ne referment pas l'élève sur la machine ; leur usage se conçoit individuellement ou par petit groupe, l'activité menée pouvant s'intégrer à d'autres.

L'élève est en mode commande. Il peut exprimer sa résolution du problème avec les moyens logiciels.

Son rapport à la machine est de la vaincre en la faisant produire.

De tels logiciels autorisent la définition de situations-problèmes, puisque non-déterministes et offrant des moyens d'expression, permettant des acquisitions dans le cadre d'une dialectique outil-objet.

A fortiori, ils autorisent une épreuve de savoirs dépassant la simple répétition. Ce qui précède suppose, outre la pertinence du logiciel, l'adéquation de la situation proposée avec lui : l'enseignant a là un rôle fondamental à jouer. Remarquons d'ailleurs que les logiciels C sont également disponibles pour des interventions de l'enseignant face à la classe.

Ainsi, une première analyse pourrait conduire à privilégier sans mesure les logiciels de type C, ne serait-ce que parce qu'ils permettraient une approche de savoirs par l'action.

Toutefois, un examen plus poussé fait apparaître des limites, que nous pensons être, comme en type A, liées à la communication.

Les tâches de communication sont particulièrement importantes, voire complexes : peut-être, l'élève n'est-il pas en mesure d'acquérir les connaissances de communication nécessaires à une utilisation performante pour un apprentissage mathématique ?

Peut-être, la dialectique apprentissage de la communication/apprentissage mathématique n'est-elle pas susceptible de fonctionner, si par ailleurs, certaines acquisitions mathématiques ou méthodologiques ne sont pas assurées ?

Peut-être, la communication particulière à un logiciel fait-elle obstacle à une bonne appréhension d'un concept ?

Un de nos résultats est que l'on peut reconnaître aux logiciels de type C (tout au moins, à certains de ces logiciels pour certains concepts) une fonction de cadre langage. Peut-être, celui-ci est-il inadapté à la notion, ou à l'élève ? En

particulier, les savoirs impliqués étant complexes et structurés des deux points de vue, mathématique et communication, un véritable recours au cadre langage nous semble peu vraisemblable à l'école élémentaire. Dans ces conditions, l'élève emploiera plutôt le cadre descriptif inclus dans le cadre langage.

Il nous paraît que le cadre descriptif permet à l'élève dans des limites individuelles et logicielles une intellectualisation de son action.

Mais une action attestée par une production presque toujours présente, en raison des possibilités de description peut n'être qu'un leurre. Elle n'assure pas un apprentissage, et en tout cas pas un apprentissage nécessairement d'ordre mathématique : peut-être l'activité n'a-t-elle pas statut d'activité mathématique, se situe-t-elle au niveau de la production d'un simple graphisme ?

Enfin, la validation dont l'élève se sent maître, n'est souvent, en l'absence de connaissances, qu'une évaluation non raisonnée.

C'est pourquoi, bien que ces logiciels nous paraissent riches de potentialités, nous ne pensons pas qu'ils se suffisent pas à eux-mêmes ; il faut en assurer une bonne gestion didactique : c'est le rôle de l'enseignant.

L'élève pratiquera alors avec eux des mathématiques, que nous qualifierons d'expérimentales.

Il se préparera à l'institutionnalisation de notions nouvelles en produisant un objet mathématiquement exact ou inexact.

Il éprouvera ses savoirs autrement que par une simple répétition.

Il s'affrontera à des situations de communication (au sens usuel) mieux contrôlées pour certaines de leurs phases, à des restrictions de choix de procédures pour résoudre un problème, à des lectures singulières de situations.

Les interventions de l'enseignant relatives à son activité, d'autres activités, avec ou sans ordinateur, participeront à l'apprentissage.

Une prospective de construction de logiciels de type C peut porter sur la définition de systèmes d'instructions de production (et de la syntaxe), adaptés à l'école élémentaire.

3d Les logiciels de type B

Le type B de logiciels est globalement analogue au type C.

Il s'en distingue toutefois par des critères qui nous intéressent :

- Les instructions couvrent souvent un moins grand domaine sémantique.

Pour cette raison, elles sont peut-être plus aptes à focaliser sur des objets mathématiques.

– La syntaxe est rudimentaire, voire inexistante : les instructions de parcours en tiennent lieu.

La présence des instructions de parcours peut s'avérer pertinente, si l'on souhaite régir le cheminement de l'élève. Dans d'autres cas, il pourrait être souhaitable qu'elles disparaissent, dans la mesure où elles interrompent le discours de l'utilisateur, ou l'organise déjà trop ; toutefois, il conviendrait que la syntaxe reste élémentaire : lorsque la tâche de communication se trouve réduite, l'élève peut exercer plus directement sur l'objet mathématique.

– L'activité est parfois définie, voire évaluée, ou susceptible de validation locale par l'élève grâce à des éléments logiciels spécifiques.

En raison de la présence des instructions de parcours, les logiciels de type B ne peuvent guère avoir statut de cadre langage.

Par contre, pour les raisons que nous venons de rappeler, ils peuvent constituer des cadres descriptifs pointus, performants.

4– UNE PROSPECTIVE DE CONSTRUCTION DE LOGICIELS

Une prospective de construction de logiciels qui nous intéresse est celle d'un développement de logiciels de type mixte B/C, c'est-à-dire, de mini-interpréteurs spécialisés sur des notions mathématiques.

Les instructions concerneraient un domaine réduit ; la syntaxe serait élémentaire, mais sans instructions de parcours ; un mode différé pourrait exister. Il s'agirait en quelque sorte de réaliser dans le domaine de l'informatique-outil pour l'enseignement, au moins ce qui se fait dans le domaine de l'informatique-outil pour la société.

Une autre direction retient aussi vivement notre attention : celle de la conception de logiciels à sémantique déclarative, en particulier pour l'exploration par l'élève des relations qu'entretiennent les objets du domaine numérique.

REFERENCES



REFERENCES BIBLIOGRAPHIQUES

- BALACHEFF Nicolas. — "Preuve et démonstration en mathématiques au collège". *Recherches en Didactique des Mathématiques*. Grenoble, La pensée sauvage, Vol. 3, n°3, 1982, p. 261-304.
- BALACHEFF Nicolas, LABORDE Colette, MEJIAS B. Problématique et genèse du concept d'itération. Rapport de recherche, IMAG, Grenoble, preprint 1985.
- BERDONNEAU Catherine. — *Recueil de pratiques pédagogiques autour de LOGO*. Paris, Institut National de Recherche Pédagogique, 1984.
- BOISNARD Danièle. — "Bruyère, banque d'exercices de mathématiques". *Rapport d'activités 1984-1985 du GRECO DIDACTIQUE, Thème 3 Informatique, Sous-thème 2 Informatique et ingénierie didactique*. Diffusion IREM de Rennes, 1985, p.121-138.
- BOSSUET Gérard. — *L'ordinateur à l'école*. Paris, PUF, 1982.
- BROUSSEAU Guy. — *Apports de l'informatique à l'enseignement des mathématiques, à la didactique, et à la formation des maîtres*. 1984.
- BUTLEN Denis. — *Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire*. Thèse de 3^e cycle, spécialité Didactique des mathématiques, Université Paris VII, 1985. Diffusion IREM Paris VII.
- BUTLEN Denis, LETHIELLEUX Claire. — "Utilisation de l'ordinateur pour l'apprentissage d'un algorithme de calcul des produits". *Cahier de Didactique des Mathématiques* n° 25-1 et n°25-2. Paris, IREM Paris VII.
- DUCEAU Eric et DOE Christophe. — *Dictionnaire Micro-informatique*. Paris, Cedic/Nathan, 1984.
- DOUADY Régine. — "Mathématiques (didactiques des)". *Encyclopédie Universalis*. Paris, 1984, p. 885 - 889.
- DOUADY Régine. — "De la didactique des mathématiques à l'heure actuelle". *Cahier de Didactique des Mathématiques* n° 6. Paris, IREM Paris VII.
- DOUADY Régine. — "Rapport enseignement apprentissage : dialectique outil-objet, jeux de cadres". *Cahier de Didactique des Mathématiques* n° 3. Paris, IREM Paris VII.
- DREYFUS Hubert L. — *Intelligence artificielle : mythes et limites*. Paris, Flammarion, 1984.
- GIANNESINI F., KANOUI H., PASERO R., VAN CANEGHEM M. — *Prolog*. Paris, InterEditions, 1985.

- GIORDAN A., NIDEGGER C., ZIMMERMAN J.L. — "Logo deux ans après : possibilités et obstacles à propos de l'utilisation d'un système Logo". *Education et Informatique*. Cedic/Nathan, n° 22 p. 31-34 et n°23 p. 40-43, 1984.
- KAY Alan. — "Les logiciels". *Pour la Science*, Novembre 1984, p. 14 - 22.
- LORIMY Bernard. — *L'informatique, mode d'emploi*. Paris, Fayard, 1985.
- MENDELSON Patrick. — "L'analyse psychologique des activités de programmation chez l'enfant". *Enfance* n° 2/3, 1985, p. 213-221.
- MENDELSON Patrick. — "Activation de schèmes de programmation et mémorisation de figures géométriques". *Journal européen de psychologie de l'éducation*. Vol 1 n° 2, 1986.
- PAPERT Seymour. — *Jaillissement de l'esprit : ordinateurs et apprentissage*. Paris, Flammarion, 1981.
- ROBERT Aline. — *Acquisition de la notion de convergence des suites numériques dans l'enseignement des mathématiques*. Thèse d'état Université Paris VII, spécialité didactique des mathématiques, 1982. Diffusion IREM Paris VII.
- ROBERT Aline. — "Acquisition de la notion de convergence". *Recherches en Didactique des Mathématiques*. Grenoble, La pensée sauvage, Vol. 3, n°3, 1982, p. 305-341.
- ROBERT Aline. — "Ingénierie didactique sur les suites numériques après le baccalauréat". *Cahier de Didactique des Mathématiques* n° 4. Paris, IREM Paris VII.
- ROGALSKI Janine. — "Alphabétisation informatique, problèmes conceptuels et didactique". *Bulletin APMEP*, n°347, 1985, p. 61-74.
- ROGALSKI Janine. — *Sur une séquence didactique en informatique*. Séminaire de didactique des mathématiques et de l'informatique, IMAG, Grenoble, 1983.
- ROGALSKI Janine. "Quelques éléments de théorie piagétienne et didactique des mathématiques". *Cahier de Didactique des Mathématiques* n° 2. Paris, IREM Paris VII.
- ROUCHIER André. — "Logo et les contenus d'enseignement et de formation." Actes du premier Colloque LOGO, Clermont-Ferrand, 1982, p. 28-44. Edité par l'Institut de Recherche sur l'Enseignement des Mathématiques d'Orléans, 1983.
- ROUCHIER André et SAMURCAY Rénan. — *Concepts informatiques et programmation. Une première analyse en classe de seconde des lycées*. Rapport de recherche, 1985. Diffusion IREM d'Orléans.
- SCHWARTZ Bertrand. — *L'informatique et l'éducation*. Paris, Documentation Française, 1981.
- SIMON Jean-Claude. — *L'éducation et l'informatisation de la société. Rapport à M. le Président de la République*. Paris, Fayard, 1981.

VERGNAUD Gérard. — "Quelques orientations théoriques et méthodologiques des recherches françaises en didactique des mathématiques." *Recherches en Didactique des Mathématiques*. Grenoble, La pensée sauvage, Vol. 2, n°2, 1981, p. 215-266.

VERGNAUD Gérard. — *L'enfant, la mathématique et la réalité*. Berne, P. Lang, 1981.

- *Imagiciels enseignement des mathématiques illustré par ordinateur*. Ouvrage collectif, Paris, Institut National de Recherche Pédagogique, 1984.
- "Quelques concepts, quelques généralités et quelques références". Collectif. *Cahier de didactique des mathématiques n°5*. Paris, IREM Paris VII.
- *Catalogue des logiciels Informatique Pour Tous*. Niort, CAMIF, 1985.
- Revue *Pour la Science*, Numéro spécial Informatique, Novembre 1984.

BIBLIOGRAPHIE SOMMAIRE

- ABELSON Harold. — *Le Logo sur APPLE*. Paris, Cedic/Nathan, 1984.
- ARSAC Jacques. — *Premières leçons de programmation*. Paris, Cedic/Nathan, 1980.
- ARSAC Jacques. — *Les bases de la programmation*. Paris, Dunod, 1983.
- BESTOUGEFF H., FARGETTE J. P. — *Enseignement et ordinateur*. Paris, Cedic/Nathan, 1982.
- BONNET Alain. — *L'intelligence artificielle. Promesses et Réalités*. Paris, InterEditions, 1984.
- BREMOND G. — *La révolution informatique*. Paris, Hatier, 1983.
- BRIAND Joël. — *Situation didactique et logiciel d'enseignement*. Mémoire de D. E. A de didactique des mathématiques, Université de Bordeaux I, 1985. Diffusion IREM de Bordeaux.
- CHAUVIN Jeanine, EIMERL Kamila. — *Le jeune enfant et le micro-ordinateur*. Paris, Documentation Française, 1984.
- CHEVALLARD Yves. — *La transposition didactique*. Grenoble, La pensée sauvage, 1985.
- FAVRE-NICOLIN Raymond, MARININCHI Jean-Baptiste. — *Tableur et pédagogie de l'informatique*. Grenoble, CRDP, 1986.
- GRAS Régis. — *Vers un enseignement par objectifs en mathématiques*. IREM de Rennes.
- HERMANT Corinne. — *Enseigner, apprendre avec l'ordinateur*. Paris, Cedic/Nathan, 1985.
- LEDGARD Henry F. — *Proverbes de programmation*. Traduit et annoté par Jacques ARSAC. Paris, Dunod, 1983.
- LEFEVRE J. M. — *Guide pratique de l'E. A. O.* Paris, Cedic/Nathan, 1984.
- NORA Simon, MINC Alain. — *L'informatisation de la société. Rapport à Monsieur le Président de la République*. Paris, La Documentation Française, Points Seuil, 1978.
- ROUCHIER André. — "Learning recursive calls in building up LOGO procedures". *Psychology of Mathematics Education* n°10. Londres, 1986, p. 289-294.
- SKINNER B. F. — *La révolution scientifique de l'enseignement*. Bruxelles, Mardaga, 1969.

THEBAUT Michel. — "Premiers éléments pour une analyse formelle des logiciels didactiques". *Education Télématique Informatique*, n° 2. Paris, LISH/CNRS, 1984, p. 63-73.

TURKLE Sherry. — *Les enfants de l'ordinateur*. Paris, Denoël, 1986.

WEIZENBAUM Joseph. — *Puissance de l'Ordinateur et Raison de l'Homme*. Paris, Editions d'Informatique, 1981.

— *Documentation des logiciels Nathan Ecoles ; Français/Mathématiques*. Paris, Cedic/nathan, 1985.

— Revue *Education et Informatique*. Paris, Cedic/Nathan.

— *Informatique et enseignement*. Actes du Colloque. Paris, Documentation française et CNDP, 1983.

ANNEXES



LISTE DES ANNEXES

- | | |
|-------------------|---|
| <u>Annexe I</u> | Descriptif sommaire des logiciels retenus pour l'établissement de la typologie a priori. |
| <u>Annexe II</u> | Quelques grilles d'analyse de logiciels. |
| <u>Annexe III</u> | Analyse a priori selon les grilles CONT et COM des logiciels retenus pour l'établissement de la typologie a priori. |
| <u>Annexe IV</u> | Classement de logiciels cités dans la recherche selon la typologie générale. |
| <u>Annexe V</u> | Principes de schématisation de la structure des logiciels, par rapport à la communication. |

ANNEXE I

Descriptif sommaire des neuf logiciels retenus pour l'établissement de la typologie a priori

PUCE

Ce logiciel a été réalisé sous notre direction par des élèves-institutrices en formation initiale.

Il s'inspire d'une activité proposée par la brochure *Aides pédagogiques pour le cours élémentaire* (publication APMEP), p. 67-68, comme approche de la division euclidienne, d'après un article de J. PAINCHAULT dans la revue *Grand N* n°13 (CRDP de Grenoble). On retrouve une description de cette activité sous le titre "Jeu de pièges" dans *ERMEL Cycle Moyen tome 1* (Paris, Hatier, 1981), p.200-201, avec un objectif de manipulation d'écritures à propos de l'apprentissage de la division euclidienne au CM1.

Le logiciel PUCE demande de traiter plusieurs situations que nous résumerons ainsi :

Une puce part d'une case de départ «d», et saute de «s» en «s». Placer une piège «p» pour l'attraper dans la zone comprise entre «p1» et «p2».

Il s'agit donc de trouver un nombre entier p congru à d modulo s et appartenant à l'intervalle $[p1 \ p2]$.

L'enfant doit traiter un cursus de problèmes.

Des "aides" optionnelles sont à sa disposition sur une partie du cursus :

– il peut visualiser une «puce graphique», sautant sur une droite, mais les nombres atteints ne s'affichent pas ; cette "aide" n'est disponible qu'en début de cursus.

– il peut utiliser une «calculatrice», incluse dans le logiciel (deux signes opératoires au maximum ; parenthèses non admises) ; cette "aide" est disponible pour un nombre de calculs déterminé pour chaque problème, ce nombre allant en décroissant sur l'ensemble du cursus.

Après deux réponses inexactes à un problème, on propose à l'élève un problème "plus facile" ; en cas de nouvel échec, le cursus est terminé.

Il y a infirmation ou confirmation des réponses de l'élève ; la solution

n'est pas donnée en cas d'échec.

Le parcours du logiciel se présente complètement sous forme de question/réponse, même l'accès aux options «Puce graphique» et «calculatrice» ; par exemple, l'accès à la calculatrice est obtenu en répondant à la question : "Veux-tu te servir de la calculatrice ? (oui/non)" ; "Donne ton calcul" ; "Veux-tu faire un autre calcul ?"

Signalons enfin, qu'il existe une erreur d'algorithme qui fait que le logiciel ne répond pas exactement à la description ci-dessus, en particulier pour la hiérarchie des situations successivement proposées ; un blocage du logiciel peut intervenir.

SIMUL ET RECT

Les logiciels SIMUL et RECT ont été conçus par Denis BUTLEN et nous-même, dans le cadre d'une recherche portant sur l'utilisation de l'ordinateur dans l'enseignement de la multiplication au cycle élémentaire.

Un descriptif détaillé et les résultats de l'expérimentation pourront être consultés dans *Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire*, par D. BUTLEN (Thèse de 3^e cycle, Université Paris VII, 1985, diffusion IREM Paris VII).

Le logiciel RECT propose plusieurs problèmes isomorphes que nous décrivons ainsi : exprimer le nombre de points d'une grille rectangulaire que l'on peut afficher, complètement ou partiellement, à l'aide de diverses instructions ; le problème posé est par conséquent un problème numérique, présenté dans un jeu de cadres numérique/graphique.

Le temps de réponse est limité.

Il n'y a ni infirmation, ni confirmation, ni solution suite à une réponse de l'élève.

Le logiciel SIMUL a été réalisé à fin d'apprentissage des instructions d'affichage disponibles dans RECT.

Sans faire intervenir de questions de dénombrement, il propose les mêmes instructions que RECT pour afficher une grille de points recouvrant l'écran ; le problème est donc d'ordre graphique.

Plusieurs instructions sont disponibles :

– Instructions de maîtrise du type d'affichage.

L'élève peut demander un affichage par ligne L, par colonne C, d'un seul coup T.

– Instructions de maîtrise du déroulement de l'affichage.

L'élève peut ne pas intervenir sur le déroulement de l'affichage en optant pour P ou A : avec P, l'affichage se réalise ligne par ligne (ou colonne par colonne), sans marquer d'arrêt au premier élément de chaque ligne (ou colonne), tandis qu'avec A un arrêt est observé.

Il peut également gérer lui-même le déroulement de l'affichage en choisissant M (Moi) : dans ce cas, en appuyant sur la barre d'espacement, il affiche le reste de la ligne (ou de la colonne) en cours, dont seul le premier élément apparaissait ; en donnant l'instruction C (respectivement L), il modifie le type d'affichage en cours de réalisation.

– Instructions concernant la modalité d'affichage d'une nouvelle grille.

L'appui de la barre d'espacement reconduit pour une nouvelle grille la procédure d'affichage définie pour la grille achevée.

L'instruction N permet de définir, avec les précédentes instructions, une nouvelle modalité.

TACHE

Ce logiciel a été conçu par Denis Butlen dans le cadre de la poursuite de sa recherche concernant la désignation multiplicative au cycle élémentaire.

Un descriptif détaillé et les résultats de l'expérimentation de ce logiciel se trouvent dans *Apport de l'ordinateur à l'apprentissage des écritures multiplicatives au cours élémentaire*, par D. BUTLEN (Thèse de 3^e cycle, Université Paris VII, 1985, diffusion IREM Paris VII).

Le problème posé à l'élève est d'ordre numérique ; toutefois, une bonne maîtrise du cadre graphique est mise en jeu.

Il s'agit d'effectuer un dénombrement de type multiplicatif des points d'une grille partiellement masquée par une "tache".

Deux genres de situations interviennent :

– les points affichés permettent de reconstituer une ligne et une colonne complètes, et par conséquent de répondre à la question.

– les points affichés ne permettent pas de reconstituer une ligne et une

colonne complètes.

L'élève dispose d'une instruction E (Encore) qui fait apparaître des points, immédiatement à côté de ceux qui sont déjà affichés.

Cette instruction E est donc nécessaire à la résolution du deuxième genre de situation ; par ailleurs, l'élève peut l'utiliser, s'il n'arrive pas à gérer immédiatement le premier type de situation.

Une instruction R permet d'entrer la réponse au problème posé.

Celle-ci doit respecter la forme «axb» ; toutefois, une écriture canonique décimale «n» est acceptée et traitée comme «nx1».

Il y a plusieurs éléments d'évaluation :

– le score de l'élève, égal à la différence entre 10 et le nombre d'appui de la touche E ; le score optimum est indiqué.

– en cas d'erreur, l'élève obtient un affichage correspondant à sa réponse, et la réponse correcte.

DIVIS

Nous avons travaillé sur une version préliminaire du logiciel ultérieurement commercialisé par Cedic/Nathan sous le nom de DIVISION.

Le lecteur pourra trouver une description de DIVISION (proche de DIVIS) dans *Documentation des logiciels Nathan Ecoles ; Français/Mathématiques*, Paris, Cedic/nathan, 1985.

Le logiciel DIVIS demande d'effectuer des divisions euclidiennes, selon un algorithme de soustractions successives.

Une visualisation accompagne le traitement numérique : on vide un réservoir d'eau à l'aide de bouteilles, de bidons, ou de camions.

Plusieurs niveaux sont proposés :

– Niveaux 1 et 2 : on peut soustraire du dividende, en pointant à chaque fois au crayon optique, un multiple du diviseur par une puissance de dix (1, 10, 100) ; le diviseur est à un chiffre au niveau 1, à deux chiffres au niveau 2.

– Niveaux 3 et 4 : on peut soustraire du dividende, en pointant à chaque fois au crayon optique dans un ordre quelconque, un multiple du diviseur par une puissance de dix (1, 10, 100), et par l'un des neuf nombres compris entre 1 et 9 ; le diviseur est à un chiffre au niveau 3, à deux chiffres au niveau 4.

En cas d'impossibilité de soustraire, un message est affiché ("C'est trop !").

Lorsque la division est terminée, l'égalité « $a=bq+r$ » concernant les situations de division euclidienne apparaît ; en outre, il y a indication d'un score (nombre de soustractions faites par l'élève, nombre de soustractions nécessaires et suffisantes).

Dans la version définitive DIVISION, il existe des possibilités d'adaptation du logiciel par l'enseignant (choix de l'intervalle où sera choisi le diviseur).

MOSAIQUE (MOS)

Le logiciel MOSAIQUE (abréviation : MOS) a été réalisé sous notre direction par des élèves-institutrices en formation initiale.

Il permet de paver l'écran à l'aide de quatre motifs choisis parmi les cinq présentés ci-dessous :



Les parties blanches (respectivement pointillées) de chaque motif peuvent être coloriées de seize couleurs différentes.

Les quatre motifs coloriés s'organisent de la façon suivante en un grand motif :

Motif 1	Motif 2
Motif 4	Motif 3

Toutefois, le pavage se construit successivement par choix et reproduction sur tout l'écran du motif 1, puis du motif 2, etc ...

On notera que les motifs sont tels que des activités d'ordre perceptif autour des notions de symétrie-droite et symétrie-point (rotation d'un demi-tour) sont impliquées ; en outre, le concept de translation est également en cause.

A chaque étape de la construction du pavage, l'élève peut revenir sur ce qu'il vient de faire. De plus, à la fin de l'activité, il peut modifier tout ou partie de sa réalisation.

Signalons enfin que le parcours du logiciel s'effectue par questions/réponses ("Veux-tu revoir les codes ?" ; "Veux-tu changer quelque chose ?") ; une interactivité plus grande pourrait être envisagée.

FRISE

Nous avons travaillé sur une version préliminaire du logiciel ultérieurement commercialisé par Cedic/Nathan, qui diffère sensiblement de la version définitive : en particulier, l'option "étudier une frise" n'est pas disponible ; on ne peut que dessiner ou regarder une frise, ou encore quitter le logiciel.

Le lecteur trouvera une description de FRISE dans sa version définitive (proche, pour les options disponibles, du logiciel que nous avons utilisé) dans *Documentation des logiciels Nathan Ecoles ; Français/ Mathématiques*, Paris, Cedic/nathan, 1985.

Rappelons qu'il est possible de classer les dessins appelés frises selon le groupe des isométries qui les laissent globalement invariants ; de plus, on peut démontrer qu'il existe sept groupes possibles d'isométries conservant globalement une frise.

Le logiciel FRISE permet d'afficher à l'écran des frises des sept types possibles.

L'utilisateur choisit, parmi sept machines, sa "machine à friser", c'est-à-dire, d'un point de vue mathématique, un système d'isométries générateur de frises.

Ensuite, il dessine avec le crayon optique, point par point, un motif sur un pavé minimum permettant de reconstituer la frise.

Le pavé translatable construit à partir du pavé minimum, et la frise s'affichent automatiquement, au fur et à mesure, à l'écran.

Ainsi, l'utilisateur emploie un système générateur minimum (pavé minimum, système d'isométries) de frises ; la frise qu'il obtient peut admettre ce système comme système générateur minimum si, par ailleurs, le dessin qu'il effectue ne comporte pas de particularités en terme d'isométries.

Il lui est possible de regarder en diverses tailles les frises qu'il a

produites.

PICTOR

Ce logiciel est édité par VIFI/Nathan.

Les périphériques d'entrée utilisés sont le clavier (pour écrire) et le crayon optique (pour dessiner ou désigner une instruction représentée par un pictogramme).

Trois lignes de pictogrammes affichables à l'écran permettent :

- d'écrire en couleur où l'on veut sur l'écran.
- d'effectuer des dessins libres ou à caractère géométrique (à partir de segments, rectangles, cercles) en traits fins ou épais.
- de colorier des domaines fermés, en tramé ou en plein.
- de gommer.
- de déplacer une zone de l'écran par rotation de 90° ou par translation.
- d'agrandir ou de réduire une zone de l'écran.
- de revoir les diverses étapes de la construction d'un dessin.
- d'effacer l'écran, d'en modifier les couleurs du fond et du bord.
- de copier l'écran sur papier, sur cassette ou sur disquette.

Signalons que les caractéristiques des transformations géométriques disponibles ne nous semblent pas susceptibles d'être déterminées par l'élève : elles ne sont pas explicites ; pratiquement, elles s'avèrent même délicates à reconnaître.

D'autre part, la notation du rapport des homothéties possibles (-2, -3, -4, +2, +3, +4) nous paraît maladroite, les élèves n'ayant déjà que trop tendance à concevoir un modèle additif pour ces transformations.

LOGO

Nous ne décrivons pas ce logiciel que nous pensons suffisamment connu.

Le lecteur pourra, le cas échéant, consulter l'ouvrage de Seymour Papert, *Jaillissement de l'esprit : ordinateurs et apprentissage* (Paris, Flammarion, 1981) et un manuel de référence du langage.

ANNEXE II

Quelques grilles d'analyse de logiciels

Le lecteur trouvera dans les pages suivantes :

- un ensemble de grilles de l'Agence De l'Informatique (ADI).
- une grille proposée par l'association Enseignement Public et Informatique (EPI).
- une grille diffusée lors du plan Informatique Pour Tous (IPT) par le Ministère de l'Education Nationale, Mission des Technologies Nouvelles.
- un questionnaire général et une grille en provenance du Centre d'Etudes des Systèmes et Technologies Avancées (CESTA).

FDI

FICHE D'IDENTIFICATION DE DIDACTICIEL

IDENTIFICATION	
NOM DU PRODUIT :	VERSION : DATE ' ' '
DISCIPLINE :	SERIE : NUMERO :
SUJET (THEME) :	COPYRIGHT (DATE DE DEPOT) :
NIVEAU (PUBLIC VISE) :	
PRE-REQUIS OU CONNAISSANCES NECESSAIRES :	
DUREE MOYENNE :	
AUTEURS :	
DIFFUSE PAR (ADRESSE, TELEPHONE) :	
OPERATIONNEL SUR (TESTS, SUPPORTS ET FORMAT, VOIR FICHE EXPERIMENTATION)	MATERIEL(S) SYSTEME(S) D'EXPLOITATION

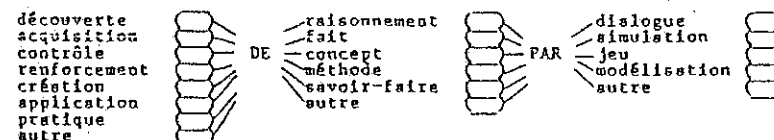
ASPECTS TECHNIQUES	
- 1 - LANGAGE(S) D'ECRITURE :	VERSION :
MAINTENANCE PREVUE JUSQU'A :	PORTABILITE :
- 2 - MEMOIRE UTILE :	
PERIPHERIQUES (marque, type) : - INDISPENSABLES	
- OPTIONNELS	
EQUIPEMENTS PARTICULIERS (Laboratoire, réseau, etc...) :	
INTERFACES :	
OPTION GRAPHIQUE :	
VIDEOTEX :	
PROBLEMES PARTICULIERS : - DU CLAVIER	
- DE L'ECRAN (lignes, colonnes, couleur)	
TEMPS DE REPONSE :	LOGICIEL OUVERT ET FERME :
- 3 - INCOMPATIBILITE EVENTUELLE :	

OBJECTIFS DECRITS :

Y-A-T-IL DES DOCUMENTS OU PROGRAMMES ANNEXES PERMETTANT D'EVALUER LES OBJECTIFS * OUI NON

CHOIX DIDACTIQUES SOUS JACENTS EVENTUELS :

ASPECTS DOMINANTS DU LOGICIEL : (chiffrer les différents chemins choisis)

DEGRE D'INTERACTIVITE :
(nombre moyen d'interactions par page-écran)QUALITE DE LA PRESENTATION :
(définition de l'image, animation, nombre de couleurs, son, etc...)

ENVIRONNEMENT PEDAGOGIQUE :

Place dans le cursus : illustration extrapolation introduction
nouvelle approche approfondissement autres

Documents d'accompagnement indispensables : conseillés :

DEROULEMENT

DUREE MOYENNE TOTALE :

CONDITIONS D'UTILISATION PREVUE : Par l'enseignant devant la classe Par l'apprenant - libre service
inséré en cours
en soutien NECESSITE-T-IL UN TRAVAIL EN EQUIPE ? OUI NON *

	Activité de l'apprenant	Activité de l'enseignant
avant		
pendant		
après		

INITIATIVES DE L'APPRENANT : facilite de progression avant arrière Choix laissé : sujet méthode niveau demande d'aide
rappel de cours demande de documentation ANALYSE DE REPONSE : OUI NON *choix multiples réponses ouvertes tolérance orthographique
numérique simple numérique ouverte (facile 2, 2/3...) formulaire (U=RI...)

SUIVI

POSSIBILITE DE GARDER UNE TRACE DU TRAVAIL :

	Pendant le déroulement	Après	Jamais <input type="checkbox"/>
Sous quelle forme ?			
A quelle fin ?			

EXPERIMENTATION

Nombre d'apprenant : Existe-t-il des documents d'expérimentation OUI NON *

(*) rayer la mention inutile

DU LOGICIEL	PAR
Intitulé.....	Monsieur ou Madame.....
Matériel utilisé (format support).....	Etablissement.....
.....
Périphérique.....	Téléphone.....
.....	Adresse personnelle.....
Documents.....
.....	Téléphone.....
Pré-requis.....	Nombre total d'apprenant.....
.....	Nombre total de passages.....

CONDITIONS DE L'EXPERIMENTATION

OBJECTIFS DE L'EXPERIMENTATION	NIVEAU	DUREE MOYENNE PAR APPRENANT	DUREE DE LA SEANCE	REMARQUES EVENTUELLES

UTILISATION

- * Facilité d'utilisation par l'apprenant ?
- * Adéquation des consignes d'utilisation :
- * Caractéristiques du dialogue :
 - Souplesse et contraintes des réponses de l'apprenant (tolérance orthographique voire syntaxique, pertinence du codage, degré de concision, etc...) :
 - Possibilité de reconnaissance de la réponse de l'apprenant :
 - . type et nombre de réponses non reconnues :
 - . adéquation aux prévisions de l'auteur :
 - . influence de la réponse sur la suite :
 - . adaptation de la progression au niveau de l'apprenant :
- * Initiatives de l'apprenant
 - . sait-il toujours ce que l'auteur attend de lui ?
 - . sait-il toujours ce qu'il peut faire ?
 - . facilité de progression à l'initiative de l'apprenant (avant, arrière) :
 - . utilisation effective des aides prévues (documentation, etc...) :

* Quelles sont vos suggestions à propos :

- . du contenu scientifique ?
- . de l'intérêt du sujet traité ?
- . de la précision des pré-requis ?
- . de la présentation ?
- . de la documentation ?
- . de la pertinence des questions ?
- . autres ?

PROBLEMES TECHNIQUES

* Difficulté de mise en place de l'expérimentation

* Problèmes techniques rencontrés ("Bugg", manipulation imprévue, erreur bloquante, etc)

* Qualités et caractéristiques du mode d'emploi du logiciel

* Avez-vous des remarques sur :

- . la mise en page ?
- . la lisibilité et la clarté des messages ?
- . le graphisme, graphique ?
- . la couleur ?
- . le son ?

* Est-il prévu de pouvoir adapter le logiciel à votre pratique ?

- . l'avez-vous fait simplement ?
- . difficultés rencontrées éventuellement :

CONCLUSION

* Adéquation aux caractéristiques pédagogiques, utilisation, limites, extension, etc..., du logiciel :

ASSOCIATION E.P.I.
ENSEIGNEMENT PÉDAGOGIQUE
1, AV. PIERRE-CORNILLE
78170 LA CELLE-SAINT-CLOUD

4- Les objectifs suivants vous satisfont-ils ?

- totalement
 en partie
 pas du tout

78170 LA CELLE-SAINT-CLOUD

* Restent-ils des traces écrites de travaux réalisés ? OUI
 NON

Commentaire: _____

* Les travaux réalisés ? satisfaisants
 en équipe

Commentaire: _____

5- Pouvez-vous décrire en quelques lignes (à moins que vous ne souhaitez rajouter un texte plus complet qui pourrait être publié) le projet (ou les démarches) pédagogiques dans le cadre duquel vous avez utilisé ce logiciel en situation réelle avec les élèves:

6- L'ordinateur apporte-t-il quelque chose de plus par rapport à une approche "traditionnelle" ? OUI
 NON

Commentaire: _____

0 0
M I. LE TRAVAIL DE L'ÉLÈVE:

* Durée (en heures) ?

Minutes: _____

Nombre: _____

Commentaire: _____

* L'utilisation en auto-serveur est-elle possible ? OUI
 NON

(Logiciel suffisamment complexe)

Commentaire: _____

* Utilisation dans le cadre de cours ou de TP en présence de l'enseignant: OUI
 NON

Commentaire: _____

* Y-a-t-il un document d'accompagnement pour l'élève ? OUI
 NON

Commentaire: _____

page 3

Votre avis sur un logiciel
APRÈS son utilisation
avec les élèves.

TITRE: _____

Le sujet: _____

Auteur(s): _____
Organisme / Éditeur: _____
Prix R.T.T.: _____

Ordre(s) écrit(s): _____
Support (disquette, cassette, ...): _____
Présentation (schémas, ...): _____
Style graphique: _____
Langage: _____
Documentation (autodocument, brochures, ...): _____

2- RAYONNAGE D'UTILISATION (selon votre propre expérience): _____

11. TYPE DE LOGICIEL:

- Apprentissage
 Renforcement des connaissances
 Contrôle Aide contrôlée
 Simulation
 Banque de données
 Jeu
 Autre: _____

ASSOCIATION E.P.I.
ENSEIGNEMENT PÉDAGOGIQUE
1, AV. PIERRE-CORNILLE
78170 LA CELLE-SAINT-CLOUD

-> Vous pouvez évidemment cocher plusieurs cases pour un même logiciel.

page 3

* Ce travail sollicite de l'élève: OUI
 NON

actif
 initiatrice
 recherche
 raisonnement (logique)
 déduction

* L'emploi se révèle-t-il: OUI
 facile
 difficile
 autre: _____

Vous pouvez
cocher
plusieurs cases

* Les élèves se sont-ils montrés intéressés ? OUI
 NON

Commentaire: _____

* Ont-ils tiré un profit évident de l'utilisation de ce logiciel ? OUI
 NON

Commentaire: _____

* Voyez-vous quelque chose d'important à rajouter sur ce chapitre ? _____

1. CONVIABILITÉ ET BÉNÉVOLENCE:

6- Message affiché: _____

* Au point de vue de la forme, les messages affichés sont-ils: OUI
- bien présentés ? NON

Commentaire: _____

- longs
 trop longs
 courts
 trop courts
 corrects

Commentaire: _____

page 4

III. STRATÉGIE PÉDAGOGIQUE:

4- Les objectifs suivants ont-ils (été) concernés (s) du logiciel:

* Sont-ils clairement précisés ? OUI
- SI OUI, où ? _____
 NON

* SI OUI, à l'intention de qui ? des élèves
 des enseignants

* Répondez ces objectifs: _____

* Sont-ils atteints ? (éventuellement dans quelle proportion ?) _____

* Le logiciel permet-il le contrôle de ces objectifs ? _____

7- Les connaissances préalables requises ("pré-requis"):

Sont-elles ? précises OUI NON
 vagues importantes

Commentaire: _____

8- LA PROGRESSION:

* Y-a-t-il des étapes pédagogiques à l'intérieur du logiciel ? OUI
 NON

Commentaire: _____

* Si oui, y-a-t-il formation des ergus de l'élève après chaque étape ? OUI
 NON

Commentaire: _____

* Y-a-t-il possibilité de choix du niveau de difficulté ? OUI
 NON

SI oui, de quelle façon ? _____

page 2

E.P.I.

CESTA

NOM

QUESTIONNAIRE GENERAL

Q 1.

Voici une liste non exhaustive de critères qui permettent d'évaluer un didacticiel.
Pour chacun de ces critères indiquez une note entre 1 et 5 où :

1 = Pas important du tout

2 = Peu important

3 = Moyennement important

4 = Important

5 = Essentiel

a) ENVIRONNEMENT D'UN DIDACTICIEL

Qualité des documents d'accompagnement	1	2	3	4	5
Pré-requis explicites	1	2	3	4	5
Exemples d'utilisation ou d'intégration	1	2	3	4	5
Public cible précisé	1	2	3	4	5
Clarté des objectifs pédagogiques	1	2	3	4	5
Durée du didacticiel	1	2	3	4	5
Facilité de mise en route	1	2	3	4	5
Possibilité de copies d'écrans	1	2	3	4	5
Autres (précisez)	1	2	3	4	5

b) CONCEPTION ET PRESENTATION DES ECRANS

Couleur	1	2	3	4	5
Son	1	2	3	4	5
Lisibilité du texte	1	2	3	4	5
Esthétique, qualité graphique	1	2	3	4	5
Jeux de caractères variés	1	2	3	4	5
Effets d'animation	1	2	3	4	5
Effets spéciaux (clignotement, sur-intensité, inversion vidéo)	1	2	3	4	5
Division de l'écran en "fenêtres"	1	2	3	4	5
Variété des pages	1	2	3	4	5
Vitesse de succession des pages-écrans, suite à une réponse de l'apprenant	1	2	3	4	5
Effacement partiel de l'écran	1	2	3	4	5
Superposition d'écrans	1	2	3	4	5
Densité des pages-écrans	1	2	3	4	5
Réponse acceptée n'importe où sur l'écran	1	2	3	4	5
Autres (précisez)	1	2	3	4	5

c) INTERACTIVITE, ASPECTS FORMELS DU DIALOGUE

Ecran tactile	1	2	3	4	5
Crayon optique	1	2	3	4	5
Souris	1	2	3	4	5
Pilotage d'équipements audiovisuels : (magnétophone, projecteur de diapos, magnétoscope, lecteur vidéodisc...)	1	2	3	4	5
Choix de la vitesse de progression	1	2	3	4	5
Retour arrière	1	2	3	4	5
Demande d'aide ou requête possible	1	2	3	4	5
Interruption de session avec restauration du contexte lors de la reprise	1	2	3	4	5
Ton humoristique	1	2	3	4	5
Caractère ludique	1	2	3	4	5
Autres (précisez)	1	2	3	4	5

d) DIDACTICIEL / STRUCTURE GLOBALE

Séquences linéaires	1	2	3	4	5
Séquences ramifiées	1	2	3	4	5
Multiniveau :					
- définition volontaire du niveau par l'enseignant	1	2	3	4	5
- définition volontaire du niveau par l'apprenant	1	2	3	4	5
- Auto-adaptation du système au niveau de l'apprenant	1	2	3	4	5
Enregistrement d'informations pédagogiques (reporting) :					
- temps de réponse de l'apprenant	1	2	3	4	5
- Arborescence parcourue	1	2	3	4	5
- Conservation des réponses	1	2	3	4	5
Possibilité d'amélioration du didacticiel	1	2	3	4	5
Suivi des élèves	1	2	3	4	5
Evaluation normative (notation classique)	1	2	3	4	5
Evaluation formative (enregistrement des progrès)	1	2	3	4	5
Evaluation sommative (pondération sur un groupe de référence)	1	2	3	4	5
Autres (précisez)	1	2	3	4	5

e) DIDACTICIEL / FONCTIONNEMENT

Q C M	1	2	3	4	5
Exercices à trous	1	2	3	4	5
Tolérances orthographiques	1	2	3	4	5
Plages de valeur (borne ou pourcentage)	1	2	3	4	5
Réponses libres avec équivalences	1	2	3	4	5
Simulation	1	2	3	4	5
Autres (précisez)	1	2	3	4	5

f) CONFORT D'UTILISATION DES MACHINES

Vous avez consulté ces didacticiels sur plusieurs machines, pour chacune des machines sur lesquelles vous avez travaillé, veuillez noter de 1 à 5 (1 = très faible, 5 = excellent), le confort d'utilisation

Apple II	1	2	3	4	5
Atari 800	1	2	3	4	5
Micral 80-22 G	1	2	3	4	5
Commodore Vic 20	1	2	3	4	5
Commodore Vic 64	1	2	3	4	5
Micro-Plato	1	2	3	4	5
HL 85	1	2	3	4	5
IBM PC	1	2	3	4	5
Silz 2	1	2	3	4	5
Logabax 529	1	2	3	4	5
Alice	1	2	3	4	5
Goupil 2	1	2	3	4	5
Goupil 3	1	2	3	4	5
Texas Instrument 99/4A	1	2	3	4	5
Thomson TO 7	1	2	3	4	5
Thomson TO 70	1	2	3	4	5
Thomson MO 5	1	2	3	4	5
Victor S 1	1	2	3	4	5

BBC Micro	1	2	3	4	5
Sinclair ZX 81	1	2	3	4	5
Sinclair Spectrum	1	2	3	4	5

Q 2.

Y a-t-il, selon vous, des matières qui ne peuvent être traitées au moyen de l'EAO ?

Q 3.

Y a-t-il, selon vous, des publics qui ne peuvent être enseignés par l'EAO ?

Q 4.

Y a-t-il, selon vous, une durée optimale d'un didacticiel, pour une session, selon les niveaux ?

	- de 20 mn	20 mn	1 h	2 h	+
Primaire					
Secondaire 1er cycle					
Secondaire 2nd cycle					
Enseignement technique					
Université					
Formation professionnelle					

Q 5.

Selon vous, l'apprentissage au moyen d'EAO doit-il s'effectuer :

- Individuellement oui non
- En petits groupes de 2 ou 3 oui non
- Plus de 3 oui non

Q 6.

Si vous deviez utiliser l'EAO (ou si vous l'utilisez déjà), pouvez-vous indiquer comment vous concevez de vous en servir ?

- Comme auxiliaire pédagogique oui non
- Comme cours oui non
- Comme contrôle oui non
- Comme approfondissement oui non
- Comme rattrapage oui non
- Comme entraînement oui non
- Comme substitution à l'acte d'enseigner oui non
- Comme outil de découverte oui non
- Comme gadget pour détendre oui non
- Pour familiariser avec les machines oui non
- Pour le raisonnement oui non

- Pour l'acquisition de : savoir faire oui non
- méthode oui non
- concept oui non

Q 7.

En fin de compte, quels sont pour vous les critères fondamentaux qui font la qualité d'un didacticiel (5 critères au plus, hiérarchisez-les par ordre d'importance décroissant, du plus important au moins important, SVP)

Q 8.

Concernant le catalogage des didacticiels, veuillez pour chacun des critères suivants indiquer s'il est selon vous ESSENTIEL ou ACCESSOIRE

	Essentiel	Accessoire
Titre		
Discipline		
Auteur		
Fournisseur		
Public visé		
Connaissances particulières requises		
Durée, temps moyen d'utilisation		
Support		
Configuration constructeurs		
Langage		
Date de lancement		
Commentaire technique		
Commentaire pédagogique		
Description sommaire		
Prix		

Q 8. bis

Quels autres critères, le cas échéant, souhaitez-vous voir mentionnés

	Tout à fait d'accord	Assez d'accord	Moyennement d'accord	Assez peu d'accord	Pas du tout d'accord
C'est une révolution enthousiasmante					
Ce n'est pas encore au point					
Finir l'échec scolaire					
Avec l'EAO, enseignant aujourd'hui = chômeur demain					
L'EAO va permettre à l'enseignant de mieux s'occuper des élèves					
Les enseignants ne sont pas prêts à utiliser l'EAO					
C'est très bien pour la formation appliquée et professionnelle					
Les inégalités dans l'accès au savoir seront aggravées					
Les machines suppriment le contact humain					
L'EAO permet de familiariser les élèves avec les nouvelles technologies					
Les élèves vont être programmés comme des machines					
C'est une conception nouvelle et prometteuse					
Pour les futurs élèves, ce sera l'apprentissage à la chaîne					

3 20

Q 9.

Voici quelques réflexions sur l'EAO qui nous ont été faites. Pour chacune de ces propositions, pouvez-vous nous dire si vous êtes "tout à fait d'accord", "assez d'accord", "moyennement d'accord", "assez peu d'accord", "pas du tout d'accord".

	Tout à fait d'accord	Assez d'accord	Moyennement d'accord	Assez peu d'accord	Pas du tout d'accord
C'est un moyen de se débarrasser des élèves					
Pour l'enseignant, c'est une libération					
C'est la fin de la culture générale					
C'est une nouvelle façon d'enseigner					
C'est une mode qui passera					
C'est très bien pour la formation initiale dans le Primaire et le Secondaire					
Les élèves n'en veulent pas					
Pour l'enseignant, c'est un surcroît de travail					

3 19

SIGNALÉTIQUE

NOM Prénom Sexe F
M

Profession Age

Adresse

NOM

TEST

Vous utilisez :

- l'EA0 oui non

- l'informatique oui non

Vous avez conçu ou participé à la réalisation :

- de didacticiels oui non
Si oui, avec quels outils (logiciels et matériels) ?

- de logiciels informatiques oui non
Si oui, avec quels langages sur quels matériels ?

Vous souhaitez concevoir ou participer à la réalisation :

- de didacticiels oui non

- de logiciels informatiques oui non

Disposez-vous d'ordinateurs :

- dans votre environnement professionnel oui non

- dans votre environnement familial oui non

Songez-vous à acquérir un ordinateur :

- professionnellement oui non

- dans votre famille oui non

Vous avez consulté plusieurs didacticiels, donnez les références de chacun de ces didacticiels dans l'ordre où vous les avez consultés, ainsi qu'un numéro d'ordre (D1, pour le premier didacticiel consulté, D2, pour le second, etc.).

Respectez ces numéros d'ordre pour répondre aux questions qui suivent.

a) ENVIRONNEMENT D'UN DIDACTICIEL

	D 1)	D 2)	D 3)	D 4)	D 5)	D 6)
Qualité des documents d'accompagnement	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Pré-requis explicites	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Exemples d'utilisation ou d'intégration	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Public cible précisé	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Clarté des objectifs pédagogiques	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Durée du didacticiel	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Facilité de mise en route	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Possibilité de copies d'écrans	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Autres (précisez)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

Q 1.

Pour chacun de ces didacticiels et pour chacun des critères proposés, donnez une note d'appréciation entre 1 et 5 où :

1 = Très mauvais pour ce critère

5 = excellent pour ce critère

	0 1)	0 2)	0 3)	0 4)	0 5)	0 6)
Superposition d'écrans	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Densité des pages-écrans	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Réponse acceptée n'importe où sur l'écran	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Autres (précisez)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

b) CONCEPTION ET PRESENTATION DES ECRANS

	0 1)	0 2)	0 3)	0 4)	0 5)	0 6)
Couleur	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Son	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Lisibilité du texte	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Esthétique, qualité graphique	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Jeux de caractères variés	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Effets d'animation	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Effets spéciaux (clignotement, sur-intensité, inversion vidéo)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Division de l'écran en "fenêtres"	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Variété des pages	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Vitesse de succession des pages-écrans, suite à une réponse de l'apprenant	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Effacement partiel de l'écran	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

7

D) DIDACTICIEL / STRUCTURE GLOBALE

	0 1)	0 2)	0 3)	0 4)	0 5)	0 6)
Séquences linéaires	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Séquences ramifiées	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Multi-niveau :						
- définition volontaire du niveau par l'enseignant ...	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
- définition volontaire du niveau par l'apprenant	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
- Auto-adaptation du système au niveau de l'apprenant	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Enregistrement d'informations pédagogiques (reporting)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
- temps de réponse de l'apprenant	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
- Arborescence parcourue	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
- Conservation des réponses	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Possibilité d'amélioration du didacticiel)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Suivi des élèves	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Evaluation normative (notation classique)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Evaluation formative (enregistrement des progrès)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

8

c) INTERACTIVITE, ASPECTS FORMELS DU DIALOGUE

	0 1)	0 2)	0 3)	0 4)	0 5)	0 6)
Ecran tactile	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Crayon optique	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Souris	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Pilotage d'équipements audiovisuels : (magnétophone, projecteur de diapos, magnétoscope, lecteur vidéodisc...)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Choix de la vitesse de progression	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Retour arrière	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Demande d'aide ou requête possible	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Interruption de session avec restauration du contexte lors de la reprise	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Ton humoristique	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Caractère ludique	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
Autres (précisez)	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

7
2
3

Q 2.

Pouvez-vous préciser, le cas échéant, les critères supplémentaires que vous auriez souhaité trouver dans cette grille, ainsi que les notes d'appréciation que vous attribuez à chacun des didacticiels pour ces critères ?

D 1	D 2	D 3	D 4	D 5	D 6
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

Q 3.

Pour chacun de ces didacticiels, donnez une note d'appréciation d'ensemble. (a = excellent, b = bon, c = moyen, d = médiocre, e = nul)

D 1)	a	b	c	d	e
D 2)	a	b	c	d	e
D 3)	a	b	c	d	e
D 4)	a	b	c	d	e
D 5)	a	b	c	d	e
D 6)	a	b	c	d	e

Evaluation sommative (pondération sur un groupe de référence) 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Autres (précisez) 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

e) DIDACTICIEL / FONCTIONNEMENT

Q C M 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Exercices à trous 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Tolérances orthographiques 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Plages de valeur (borne ou pourcentage) 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Réponses libres avec équivalences 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Simulation 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Autres (précisez) 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Q 4.

Quelle que soit la note que vous attribuez, pensez-vous que d'une manière générale, ce didacticiel peut s'intégrer utilement dans une démarche éducative ?

	dans le milieu familial	dans le milieu scolaire	en auto-formation scolaire	en auto-formation profession.	en séminaire formation profession.
D 1)	oui non	oui non	oui non	oui non	oui non
D 2)	oui non	oui non	oui non	oui non	oui non
D 3)	oui non	oui non	oui non	oui non	oui non
D 4)	oui non	oui non	oui non	oui non	oui non
D 5)	oui non	oui non	oui non	oui non	oui non
D 6)	oui non	oui non	oui non	oui non	oui non
D 7)	oui non	oui non	oui non	oui non	oui non

Q 5.

Pour chacun de ces didacticiels, indiquez le genre dans lequel vous estimez personnellement devoir le classer.

	D 1)	D 2)	D 3)	D 4)	D 5)	D 6)
Apprentissage, cours						
Entraînement, renforcement						
Approfondissement						
Remise à niveau, rattrapage						
Contrôle, évaluation, test						
Simulation						
Jeux						
Autres (précisez)						

Q 6.

Pour chacun de ces didacticiels, pouvez-vous indiquer à quel public il est, selon vous, particulièrement adapté ?

D 1)

D 2)

D 3)

D 4)

D 5)

D 6)

D 7)

Q 7.

Pour chacun de ces didacticiels, pouvez-vous brièvement indiquer ses points forts et ses points faibles ?

D 1)

D 2)

D 3)

D 4)

D 5)

D 6)

D 7)

ANNEXE III

Analyse a priori selon les grilles CONT et COM des logiciels retenus pour l'établissement de la typologie a priori

Nous présentons dans les pages suivantes l'analyse a priori selon CONT et COM de chacun des neuf logiciels PUCE, SIMUL, RECT, TACHE, DIVIS, MOSAIQUE, FRISE, PICTOR, LOGO.

Une croix indique la réponse (oui/non) à la question posée ; lorsqu'il y a lieu de nuancer, la croix se trouve entre parenthèses.

Un trait signifie que la question est sans objet.

Grille CONT		Nom du logiciel : PUCE	
Niveau concerné		Oui	Non
CP			X
CE		X	
CM		X	
Concepts mathématiques impliqués par le logiciel			
Type arithmétique		X	
Type géométrique			X
Le logiciel permet-il d'aborder plusieurs concepts de même type ?			X
Organisation du logiciel par rapport à un concept			
Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?		X	
Un nombre fini déterminé ?		X	
Une infinité théorique ?			X
Les problèmes sont-ils nécessairement isomorphes ?		X	

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

Oui Non

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?	X	
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?		X
- le complément de définition porte-t-il sur l'objet à produire ?		
- le complément de définition porte-t-il sur la démarche à tenir ?		
Y-en-a-t-il ?	X	
Sur l'écran ?	X	
Sur une documentation ?		X
Sont-elles données en une fois pour tout un problème ?	(X)	
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il suivre un cursus de plusieurs problèmes ?	X	
L'enfant doit-il répondre à des questions conceptuelles ?		X
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?	X	

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message

. écriture

. graphisme

. son

- Contenu et cas d'envoi du message

. réponse exacte au problème

- en cas d'erreur

- toujours

. information ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

Oui Non

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?	X	
La machine "fait-elle la validation" pour l'enfant ?	X	
L'enfant constate-t-il la validation de son activité par message du logiciel :	X	
- Support du message		
. écriture	X	
. graphisme		X
. son	X	
- Contenu et cas d'envoi du message		
. réponse exacte au problème		X
- en cas d'erreur		
- toujours		
. information ou confirmation de la réponse	X	
. autres éléments		X
L'enfant peut-il se sentir maître de la validation ?	X	
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?		
. Autrement ?		

Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

Sur un document ?

Sur un autre type de support ?

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

Certaines sont-elles valides sur une partie du logiciel ?

Certaines sont-elles valides à un seul endroit du logiciel ?

Oui Non

Oui	Non
X	
	X
X	
	X
	X

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

Evaluation

Caractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type I

Y-a-t-il exécution avec production d'objets commandés par l'entrée-utilisateur ?

Oui Non

Oui	Non
	X
	X
X	
	X
	X
	X
X	
	X

Effet de type 2

Y-a-t-il envoi d'une donnée logiciel ?

Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :

- message d'évaluation par rapport à l'attendu conceptuel du logiciel
- donnée d'un score
- donnée de renseignements concernant le temps
- proposition d'un outil spécifique
- proposition de visualisation du problème
- autres instructions

Nombre de types d'effets (1 ou 2)Support réservé à l'évaluation

- Graphique écran
- Support sonore
- Support mobile

Statut des entrées-utilisateur :degré de programmation laissé à l'utilisateur

Y-a-t-il présence de :

- réponse à la machine
- commandé de la machine de type pilotage
- commande de la machine de type programme

L'expression de la réponse au problème traité se fait-elle :

- en mode réponse
- en mode commande

Oui Non

Oui	Non
X	
X	
	X
	X
X	
X	
X	
1	
X	
X	
	X
X	
	X
X	
	X
X	
	X

Grille CONT

Nom du logiciel : SIMUL

Niveau concerné

CP

CE

CM

Concepts mathématiques impliqués par le logiciel

Type arithmétique

Type géométrique

Le logiciel permet-il d'aborder plusieurs concepts de même type ?

Organisation du logiciel par rapport à un concept

Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?

Un nombre fini déterminé ?

Une infinité théorique ?

Les problèmes sont-ils nécessairement isomorphes ?

Oui Non

	Oui	Non
	X	
	X	
	X	
		X
	X	
		X
	X	
	X	
		X
	X	
		X
	X	

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

Oui Non

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?	X	
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?	X	
- le complément de définition porte-t-il sur l'objet à produire ?	X	
- le complément de définition porte-t-il sur la démarche à tenir ?	X	
Y-en-a-t-il ?		X
Sur l'écran ?		
Sur une documentation ?		
Sont-elles données en une fois pour tout un problème ?		
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il suivre un cursus de plusieurs problèmes ?	X	
L'enfant doit-il répondre à des questions conceptuelles ?		X
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?		X

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message

. écriture

. graphisme

. son

- Contenu et cas d'envoi du message

. réponse exacte au problème

- en cas d'erreur

- toujours

. infirmation ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

Oui Non

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?	X	
La machine "fait-elle la validation" pour l'enfant ?		X
L'enfant constate-t-il la validation de son activité par message du logiciel :		
- Support du message		
. écriture		
. graphisme		
. son		
- Contenu et cas d'envoi du message		
. réponse exacte au problème		
- en cas d'erreur		
- toujours		
. infirmation ou confirmation de la réponse		
. autres éléments		
L'enfant peut-il se sentir maître de la validation ?	X	
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?	X	
. Autrement ?		X

Les instructions pour la réalisation de l'activité : les données du logiciel et de son environnement concernant la communication machine-utilisateur

Oui Non

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

X X
X X
X X
X

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

X X
X

Sur un document ?

X

Sur un autre type de support ?

X

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

X

Certaines sont-elles valides sur une partie du logiciel ?

X

Certaines sont-elles valides à un seul endroit du logiciel ?

X

Oui Non

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

X X
X X
X

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

X X
(X) X
X

Evaluation

Caractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

X X
X

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type I

Y-a-t-il l'exécution avec production d'objets commandés par l'entrée-utilisateur ?

X

	Oui	Non
<u>Effet de type 2</u>		
Y-a-t-il envoi d'une donnée logiciel ?	X	
Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :		
- message d'évaluation par rapport à l'attendu conceptuel du logiciel		X
- donnée d'un score		X
- donnée de renseignements concernant le temps		X
- proposition d'un outil spécifique		X
- proposition de visualisation du problème		X
- autres instructions	X	
<u>Nombre de types d'effets (1 ou 2)</u>	2	
<u>Support réservé à l'évaluation</u>		
Graphique écran	X	
Support sonore		X
Support mobile		X
<u>Statut des entrées-utilisateur :</u> <u>degré de programmation laissé à l'utilisateur</u>		
Y-a-t-il présence de :		
- réponse à la machine		X
- commande de la machine de type pilotage	X	
- commande de la machine de type programme		X
L'expression de la réponse au problème traité se fait-elle :		
- en mode réponse		X
- en mode commande	X	

Grille CONT		Nom du logiciel : RECT	
	Oui	Non	
Niveau concerné.			
CP			X
CE	X		
CM	X		
Concepts mathématiques impliqués par le logiciel			
Type arithmétique	X		
Type géométrique			X
Le logiciel permet-il d'aborder plusieurs concepts de même type ?			X
Organisation du logiciel par rapport à un concept			
Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?	X		
Un nombre fini déterminé ?	X		
Une infinité théorique ?			X
Les problèmes sont-ils nécessairement isomorphes ?	X		

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?	X	
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?		X
- le complément de définition porte-t-il sur l'objet à produire ?		
- le complément de définition porte-t-il sur la démarche à tenir ?		
Y-en-a-t-il ?		X
Sur l'écran ?		
Sur une documentation ?		
Sont-elles données en une fois pour tout un problème ?		
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il suivre un cursus de plusieurs problèmes ?	X	
L'enfant doit-il répondre à des questions conceptuelles ?		X
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?		X

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message
 - . écriture
 - . graphisme
 - . son

- Contenu et cas d'envoi du message
 - . réponse exacte au problème
 - en cas d'erreur
 - toujours

. infirmation ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?		X
La machine "fait-elle la validation" pour l'enfant ?		
L'enfant constate-t-il la validation de son activité par message du logiciel :		
- Support du message		
. écriture		
. graphisme		
. son		
- Contenu et cas d'envoi du message		
. réponse exacte au problème		
- en cas d'erreur		
- toujours		
. infirmation ou confirmation de la réponse		
. autres éléments		
L'enfant peut-il se sentir maître de la validation ?		
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?		
. Autrement ?		

Grille COM

Nom du logiciel : RECT

Les instructions pour la réalisation de l'activité
les données du logiciel et de son environnement
concernant la communication machine-utilisateur

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
b - instructions questions à l'utilisateur par rapport
au problème qu'il traite ?
c - instructions de parcours du logiciel ?
d - instructions qui permettent à l'utilisateur
de faire produire la machine ?

Support des instructionsY-a-t-il des instructions (autres qu'une invite
utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support
nécessaire ?

Sur un document ?

Sur un autre type de support ?

Validité des instructionsPour les instructions qui peuvent être des entrées
utilisateur (instructions de type d et c - branche-
ment dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

Certaines sont-elles valides sur une partie du logiciel ?

Certaines sont-elles valides à un seul endroit du logiciel ?

Oui Non

Oui	Non
	X
	X
X	
X	
X	
	X
X	
	X
X	
X	

Les entrées-utilisateur

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison
d'instructions ?

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole
d'invite ?
- sans recevoir d'instructions autres que la précédente
instruction ?
- sans provoquer d'évaluation ?

EvaluationCaractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

Effet de l'évaluationEtude de la production de la machine suite à une entrée
utilisateur valide (c'est-à-dire ne donnant pas lieu à un
message de correction syntaxique/sémantique par
rapport au discours-utilisateur) :Effet de type 1Y-a-t-il exécution avec production d'objets commandés
par l'entrée-utilisateur ?

Oui Non

Oui	Non
X	
	X
X	
	X
(X)	
	X
X	
	X
X	

Effet de type 2

Y-a-t-il envoi d'une donnée logiciel ?

Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :

- message d'évaluation par rapport à l'attendu conceptuel du logiciel
- donnée d'un score
- donnée de renseignements concernant le temps
- proposition d'un outil spécifique
- proposition de visualisation du problème
- autres instructions

Nombre de types d'effets (1 ou 2)Support réservé à l'évaluation

- Graphique écran
- Support sonore
- Support mobile

Statut des entrées-utilisateur :degré de programmation laissé à l'utilisateur

Y-a-t-il présence de :

- réponse à la machine
- commande de la machine de type pilotage
- commande de la machine de type programme

L'expression de la réponse au problème traité se fait-elle :

- en mode réponse
- en mode commande

	Oui	Non
Y-a-t-il envoi d'une donnée logiciel ?	X	
Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :		
- message d'évaluation par rapport à l'attendu conceptuel du logiciel		X
- donnée d'un score		X
- donnée de renseignements concernant le temps	X	
- proposition d'un outil spécifique		X
- proposition de visualisation du problème		X
- autres instructions	X	
<u>Nombre de types d'effets (1 ou 2)</u>	2	
<u>Support réservé à l'évaluation</u>		
Graphique écran	X	
Support sonore		X
Support mobile		X
<u>Statut des entrées-utilisateur :</u>		
<u>degré de programmation laissé à l'utilisateur</u>		
Y-a-t-il présence de :		
- réponse à la machine	X	
- commande de la machine de type pilotage	X	
- commande de la machine de type programme		X
L'expression de la réponse au problème traité se fait-elle :		
- en mode réponse	X	
- en mode commande		X

Grille CONT

Nom du logiciel : TACHE

Niveau concerné

CP

CE

CM

Concepts mathématiques impliqués par le logiciel

Type arithmétique

Type géométrique

Le logiciel permet-il d'aborder plusieurs concepts de même type ?

Organisation du logiciel par rapport à un concept

Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?

Un nombre fini déterminé ?

Une infinité théorique ?

Les problèmes sont-ils nécessairement isomorphes ?

	Oui	Non
Niveau concerné		
CP		X
CE	X	
CM	X	
Concepts mathématiques impliqués par le logiciel		
Type arithmétique	X	
Type géométrique	(X)	
Le logiciel permet-il d'aborder plusieurs concepts de même type ?		X
Organisation du logiciel par rapport à un concept		
Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?	X	
Un nombre fini déterminé ?	X	
Une infinité théorique ?		X
Les problèmes sont-ils nécessairement isomorphes ?	X	

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

Oui Non

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?	X	
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?	X	
- le complément de définition porte-t-il sur l'objet à produire ?		X
- le complément de définition porte-t-il sur la démarche à tenir ?	X	
Y-en-a-t-il ?		X
Sur l'écran ?		
Sur une documentation ?		
Sont-elles données en une fois pour tout un problème ?		
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il suivre un cursus de plusieurs problèmes ?	X	
L'enfant doit-il répondre à des questions conceptuelles ?		X
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?	X	

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message
 - . écriture
 - . graphisme
 - . son

- Contenu et cas d'envoi du message
 - . réponse exacte au problème
 - en cas d'erreur
 - toujours

. infirmation ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

Oui Non

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?	X	
La machine "fait-elle la validation" pour l'enfant ?	X	
L'enfant constate-t-il la validation de son activité par message du logiciel :	X	
- Support du message		
. écriture	X	
. graphisme	X	
. son		X
- Contenu et cas d'envoi du message		
. réponse exacte au problème	X	
- en cas d'erreur	X	
- toujours		X
. infirmation ou confirmation de la réponse	X	
. autres éléments	X	
L'enfant peut-il se sentir maître de la validation ?		X
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?		
. Autrement ?		

Grille COM

Nom du logiciel : TACHE

Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

Sur un document ?

Sur un autre type de support ?

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

Certaines sont-elles valides sur une partie du logiciel ?

Certaines sont-elles valides à un seul endroit du logiciel ?

Oui Non

Oui	Non
	X
	X
X	
X	
X	
	X
	X
X	
	X

Les entrées-utilisateur

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

EvaluationCaractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type 1

Y-a-t-il exécution avec production d'objets commandés par l'entrée-utilisateur ?

Oui Non

Oui	Non
X	
	X
X	
	X
X	
	X
X	

	Oui	Non
<u>Effet de type 2</u>		
Y-a-t-il envoi d'une donnée logiciel ?	X	
Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :		
- message d'évaluation par rapport à l'attendu conceptuel du logiciel	X	
- donnée d'un score	X	
- donnée de renseignements concernant le temps	(X)	X
- proposition d'un outil spécifique		X
- proposition de visualisation du problème		X
- autres instructions	X	
<u>Nombre de types d'effets (1 ou 2)</u>	2	
<u>Support réservé à l'évaluation</u>		
Graphique écran	X	
Support sonore		X
Support mobile		X
<u>Statut des entrées-utilisateur :</u>		
<u>degré de programmation laissé à l'utilisateur</u>		
Y-a-t-il présence de :		
- réponse à la machine	X	
- commande de la machine de type pilotage	X	
- commande de la machine de type programme		X
L'expression de la réponse au problème traité se fait-elle :		
- en mode réponse	X	
- en mode commande		X

Grille CONT		Nom du logiciel : DIVIS	
	Oui	Non	
Niveau concerné.			
CP			X
CE			X
CM	X		
Concepts mathématiques impliqués par le logiciel			
Type arithmétique	X		
Type géométrique			X
Le logiciel permet-il d'aborder plusieurs concepts de même type ?			X
Organisation du logiciel par rapport à un concept			
Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?	X		
Un nombre fini déterminé ?	X		
Une infinité théorique ?			X
Les problèmes sont-ils nécessairement isomorphes ?	X		

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

Oui Non

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?	X	
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?	X	
- le complément de définition porte-t-il sur l'objet à produire ?		X
- le complément de définition porte-t-il sur la démarche à tenir ?	X	
Y-en-a-t-il ?	X	
Sur l'écran ?	X	
Sur une documentation ?		X
Sont-elles données en une fois pour tout un problème ?	X	
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?		X
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?	X	
L'enfant doit-il suivre un cursus de plusieurs problèmes ?	X	
L'enfant doit-il répondre à des questions conceptuelles ?		X
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?		X

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message
 - . écriture
 - . graphisme
 - . son

- Contenu et cas d'envoi du message
 - . réponse exacte au problème
 - en cas d'erreur
 - toujours

. infirmation ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

Oui Non

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?	X	
La machine "fait-elle la validation" pour l'enfant ?	X	
L'enfant constate-t-il la validation de son activité par message du logiciel :	X	
- Support du message	X	
. écriture	(X)	
. graphisme	X	
. son	X	
- Contenu et cas d'envoi du message	X	
. réponse exacte au problème	X	
- en cas d'erreur	X	
- toujours	X	
. infirmation ou confirmation de la réponse	X	
. autres éléments	X	
L'enfant peut-il se sentir maître de la validation ?	(X)	
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?	X	
. Autrement ?		X

Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur

Oui Non

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

X
X
X
X

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

X
X

Sur un document ?

(X)

Sur un autre type de support ?

X

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

X

Certaines sont-elles valides sur une partie du logiciel ?

X

Certaines sont-elles valides à un seul endroit du logiciel ?

X

Les entrées-utilisateur

Oui Non

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

X
X
X

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

X
X
X

Evaluation

Caractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

X
X

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type I

Y-a-t-il exécution avec production d'objets commandés par l'entrée-utilisateur ?

X

Effet de type 2

Y-a-t-il envoi d'une donnée logiciel ?

Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :

- message d'évaluation par rapport à l'attendu conceptuel du logiciel
- donnée d'un score
- donnée de renseignements concernant le temps
- proposition d'un outil spécifique
- proposition de visualisation du problème
- autres instructions

Nombre de types d'effets (1 ou 2)Support réservé à l'évaluation

- Graphique écran
- Support sonore
- Support mobile

Statut des entrées-utilisateur :degré de programmation laissé à l'utilisateur

Y-a-t-il présence de :

- réponse à la machine
- commande de la machine de type pilotage
- commande de la machine de type programme

L'expression de la réponse au problème traité se fait-elle :

- en mode réponse
- en mode commande

	Oui	Non
Y-a-t-il envoi d'une donnée logiciel ?	X	
Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :		
- message d'évaluation par rapport à l'attendu conceptuel du logiciel	X	
- donnée d'un score	X	
- donnée de renseignements concernant le temps		X
- proposition d'un outil spécifique		X
- proposition de visualisation du problème		(X)
- autres instructions	X	
<u>Nombre de types d'effets (1 ou 2)</u>	2	
<u>Support réservé à l'évaluation</u>		
Graphique écran	X	
Support sonore	X	
Support mobile		X
<u>Statut des entrées-utilisateur :</u>		
<u>degré de programmation laissé à l'utilisateur</u>		
Y-a-t-il présence de :		
- réponse à la machine		X
- commande de la machine de type pilotage	X	
- commande de la machine de type programme		X
L'expression de la réponse au problème traité se fait-elle :		
- en mode réponse		X
- en mode commande	X	

Grille CONT

Nom du logiciel : MOSAÏQUE (MOS)

Niveau concerné

CP

CE

CM

Concepts mathématiques impliqués par le logiciel

Type arithmétique

Type géométrique

Le logiciel permet-il d'aborder plusieurs concepts de même type ?

Organisation du logiciel par rapport à un concept

Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?

Un nombre fini déterminé ?

Une infinité théorique ?

Les problèmes sont-ils nécessairement isomorphes ?

	Oui	Non
Niveau concerné		
CP	X	
CE	X	
CM	X	
Concepts mathématiques impliqués par le logiciel		
Type arithmétique		X
Type géométrique	X	
Le logiciel permet-il d'aborder plusieurs concepts de même type ?	X	
Organisation du logiciel par rapport à un concept		
Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?	X	
Un nombre fini déterminé ?	X	
Une infinité théorique ?	X	
Les problèmes sont-ils nécessairement isomorphes ?		X

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

Oui Non

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?	X	
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?	X	
- le complément de définition porte-t-il sur l'objet à produire ?	X	
- le complément de définition porte-t-il sur la démarche à tenir ?	X	
Y-en-a-t-il ?	X	
Sur l'écran ?	X	
Sur une documentation ?		X
Sont-elles données en une fois pour tout un problème ?	X	
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?	X	
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?	X	
L'enfant doit-il suivre un cursus de plusieurs problèmes ?	X	
L'enfant doit-il répondre à des questions conceptuelles ?	X	
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?	X	

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message

. écriture

. graphisme

. son

- Contenu et cas d'envoi du message

. réponse exacte au problème

- en cas d'erreur

- toujours

. infirmation ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

Oui Non

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?	X	
La machine "fait-elle la validation" pour l'enfant ?		X
L'enfant constate-t-il la validation de son activité par message du logiciel :		
- Support du message		
. écriture		
. graphisme		
. son		
- Contenu et cas d'envoi du message		
. réponse exacte au problème		
- en cas d'erreur		
- toujours		
. infirmation ou confirmation de la réponse		
. autres éléments		
L'enfant peut-il se sentir maître de la validation ?	X	
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?	X	
. Autrement ?		X

Grille COM

Nom du logiciel : MOSAÏQUE (MOS)

Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

Sur un document ?

Sur un autre type de support ?

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

Certaines sont-elles valides sur une partie du logiciel ?

Certaines sont-elles valides à un seul endroit du logiciel ?

Oui Non

Oui	Non
X	
	X
X	
X	
X	
	X
	X
	X
	X
X	
X	
	X

Les entrées-utilisateur

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

EvaluationCaractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type I

Y-a-t-il exécution avec production d'objets commandés par l'entrée-utilisateur ?

Oui Non

Oui	Non
X	
	X
(X)	
	X
(X)	
	X
X	
	X
X	

	Oui	Non
<u>Effet de type 2</u>		
Y-a-t-il envoi d'une donnée logiciel ?	X	
Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :		
- message d'évaluation par rapport à l'attendu conceptuel du logiciel		X
- donnée d'un score		X
- donnée de renseignements concernant le temps		X
- proposition d'un outil spécifique		X
- proposition de visualisation du problème		X
- autres instructions	X	
<u>Nombre de types d'effets (1 ou 2)</u>	2	
<u>Support réservé à l'évaluation</u>		
Graphique écran	X	
Support sonore		X
Support mobile		X
<u>Statut des entrées-utilisateur : degré de programmation laissé à l'utilisateur</u>		
Y-a-t-il présence de :		
- réponse à la machine	(x)	
- commande de la machine de type pilotage	X	
- commande de la machine de type programme		X
L'expression de la réponse au problème traité se fait-elle :		
- en mode réponse		X
- en mode commande	X	

Grille CONT		Nom du logiciel : FRISE	
	Oui	Non	
Niveau concerné			
CP	X		
CE	X		
CM	X		
Concepts mathématiques impliqués par le logiciel			
Type arithmétique			X
Type géométrique	X		
Le logiciel permet-il d'aborder plusieurs concepts de même type ?	X		
Organisation du logiciel par rapport à un concept			
Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?	X		
Un nombre fini déterminé ?	X		
Une infinité théorique ?	X		
Les problèmes sont-ils nécessairement isomorphes ?			X

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

Oui Non

X

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

X

- le complément de définition porte-t-il sur l'objet à produire ?

X

- le complément de définition porte-t-il sur la démarche à tenir ?

X

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

(X)

Sur l'écran ?

(X)

Sur une documentation ?

X

Sont-elles données en une fois pour tout un problème ?

(X)

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

X

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

X

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

X

L'enfant doit-il répondre à des questions conceptuelles ?

X

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

X

La validation de l'activité par rapport à son contenu

Oui Non

L'enfant peut-il valider ? Y-a-t-il un effet ?

X

La machine "fait-elle la validation" pour l'enfant ?

X

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message

- . écriture
- . graphisme
- . son

- Contenu et cas d'envoi du message

- . réponse exacte au problème
- en cas d'erreur
- toujours

. information ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

X

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

X

. Autrement ?

X

Grille COM

Nom du logiciel : FRISE

Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur

Oui Non

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
b - instructions questions à l'utilisateur par rapport
au problème qu'il traite ?
c - instructions de parcours du logiciel ?
d - instructions qui permettent à l'utilisateur
de faire produire la machine ?

(x)

X

X

X

Support des instructionsY-a-t-il des instructions (autres qu'une invite
utilisateur) :

Sur l'écran ?

X

Le périphérique d'entrée rend-il ce support
nécessaire ?

X

Sur un document ?

(x)

Sur un autre type de support ?

X

Validité des instructionsPour les instructions qui peuvent être des entrées
utilisateur (instructions de type d et c - branche-
ment dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

X

Certaines sont-elles valides sur une partie du logiciel ?

X

Certaines sont-elles valides à un seul endroit du logiciel ?

X

Les entrées-utilisateur

Oui Non

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison
d'instructions ?

X

X

X

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole
d'invite ?
- sans recevoir d'instructions autres que la précédente
instruction ?
- sans provoquer d'évaluation ?

X

X

EvaluationCaractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

X

X

Effet de l'évaluationEtude de la production de la machine suite à une entrée
utilisateur valide (c'est-à-dire ne donnant pas lieu à un
message de correction syntaxique/sémantique par
rapport au discours-utilisateur) :Effet de type 1Y-a-t-il exécution avec production d'objets commandés
par l'entrée-utilisateur ?

X

Effet de type 2

Y-a-t-il envoi d'une donnée logiciel ?

Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :

- message d'évaluation par rapport à l'attendu conceptuel du logiciel
- donnée d'un score
- donnée de renseignements concernant le temps
- proposition d'un outil spécifique
- proposition de visualisation du problème
- autres instructions

Oui Non

X

X

X

X

X

X

X

Nombre de types d'effets (1 ou 2)

2

Support réservé à l'évaluation

- Graphique écran
- Support sonore
- Support mobile

X

X

X

Statut des entrées-utilisateur :degré de programmation laissé à l'utilisateur

Y-a-t-il présence de :

- réponse à la machine
- commande de la machine de type pilotage
- commande de la machine de type programme

X

X

X

L'expression de la réponse au problème traité se fait-elle :

- en mode réponse
- en mode commande

X

X

Grille CONT

Nom du logiciel : PICTOR

Niveau concerné

CP

X

CE

X

CM

X

Concepts mathématiques impliqués par le logiciel

Type arithmétique

X

Type géométrique

X

Le logiciel permet-il d'aborder plusieurs concepts de même type ?

X

Organisation du logiciel par rapport à un concept

Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?

X

Un nombre fini déterminé ?

X

Une infinité théorique ?

X

Les problèmes sont-ils nécessairement isomorphes ?

X

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?		X
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?	X	
- le complément de définition porte-t-il sur l'objet à produire ?	X	
- le complément de définition porte-t-il sur la démarche à tenir ?	X	
Y-en-a-t-il ?		X
Sur l'écran ?		
Sur une documentation ?		
Sont-elles données en une fois pour tout un problème ?		
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?		
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?		
L'enfant doit-il suivre un cursus de plusieurs problèmes ?		
L'enfant doit-il répondre à des questions conceptuelles ?		
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?		

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message
 - . écriture
 - . graphisme
 - . son

- Contenu et cas d'envoi du message
 - . réponse exacte au problème
 - en cas d'erreur
 - toujours

. information ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?	X	
La machine "fait-elle la validation" pour l'enfant ?		X
L'enfant constate-t-il la validation de son activité par message du logiciel :		
- Support du message		
. écriture		
. graphisme		
. son		
- Contenu et cas d'envoi du message		
. réponse exacte au problème		
- en cas d'erreur		
- toujours		
. information ou confirmation de la réponse		
. autres éléments		
L'enfant peut-il se sentir maître de la validation ?	X	
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?	X	
. Autrement ?		X

Grille COM

Nom du logiciel : PICTOR

Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

Sur un document ?

Sur un autre type de support ?

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

Certaines sont-elles valides sur une partie du logiciel ?

Certaines sont-elles valides à un seul endroit du logiciel ?

Oui Non

Oui	Non
	X
	X
	X
X	
X	
	X
X	

Les entrées-utilisateur

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

EvaluationCaractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type 1

Y-a-t-il exécution avec production d'objets commandés par l'entrée-utilisateur ?

Oui Non

Oui	Non
X	
(X)	X
X	
(X)	
X	X
X	

Effet de type 2

Y-a-t-il envoi d'une donnée logiciel ?

Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :

- message d'évaluation par rapport à l'attendu conceptuel du logiciel
- donnée d'un score
- donnée de renseignements concernant le temps
- proposition d'un outil spécifique
- proposition de visualisation du problème
- autres instructions

Nombre de types d'effets (1 ou 2)Support réservé à l'évaluation

- Graphique écran
- Support sonore
- Support mobile

Statut des entrées-utilisateur :degré de programmation laissé à l'utilisateur

Y-a-t-il présence de :

- réponse à la machine
- commande de la machine de type pilotage
- commande de la machine de type programme

L'expression de la réponse au problème traité se fait-elle :

- en mode réponse
- en mode commande

Oui Non

	Oui	Non
Y-a-t-il envoi d'une donnée logiciel ?		X
Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :		
- message d'évaluation par rapport à l'attendu conceptuel du logiciel		
- donnée d'un score		
- donnée de renseignements concernant le temps		
- proposition d'un outil spécifique		
- proposition de visualisation du problème		
- autres instructions		
<u>Nombre de types d'effets (1 ou 2)</u>	1	
<u>Support réservé à l'évaluation</u>		
Graphique écran	X	
Support sonore		X
Support mobile		X
<u>Statut des entrées-utilisateur :</u>		
<u>degré de programmation laissé à l'utilisateur</u>		
Y-a-t-il présence de :		
- réponse à la machine		X
- commande de la machine de type pilotage	X	
- commande de la machine de type programme	(X)	
L'expression de la réponse au problème traité se fait-elle :		
- en mode réponse		X
- en mode commande	X	

Grille CONT

Nom du logiciel : LOGO

Niveau concerné

CP

CE

CM

Concepts mathématiques impliqués par le logiciel

Type arithmétique

Type géométrique

Le logiciel permet-il d'aborder plusieurs concepts de même type ?

Organisation du logiciel par rapport à un concept

Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?

Un nombre fini déterminé ?

Une infinité théorique ?

Les problèmes sont-ils nécessairement isomorphes ?

Oui Non

	Oui	Non
Niveau concerné		
CP	X	
CE	X	
CM	X	
<u>Concepts mathématiques impliqués par le logiciel</u>		
Type arithmétique	X	
Type géométrique	X	
Le logiciel permet-il d'aborder plusieurs concepts de même type ?	X	
<u>Organisation du logiciel par rapport à un concept</u>		
Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?	X	
Un nombre fini déterminé ?	X	
Une infinité théorique ?	X	
Les problèmes sont-ils nécessairement isomorphes ?		X

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

Oui Non

	Oui	Non
Le logiciel décide-t-il des problèmes à traiter ?		X
L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?	X	
- le complément de définition porte-t-il sur l'objet à produire ?	X	
- le complément de définition porte-t-il sur la démarche à tenir ?	X	
Y-en-a-t-il ?		X
Sur l'écran ?		
Sur une documentation ?		
Sont-elles données en une fois pour tout un problème ?		
L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?		
L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?		
L'enfant doit-il suivre un cursus de plusieurs problèmes ?		
L'enfant doit-il répondre à des questions conceptuelles ?		
L'enfant peut-il avoir recours à une aide proposée par le logiciel ?		

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message

. écriture

. graphisme

. son

- Contenu et cas d'envoi du message

. réponse exacte au problème

- en cas d'erreur

- toujours

. infirmation ou confirmation de la réponse

. autres éléments

L'enfant peut-il se sentir maître de la validation ?

. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?

. Autrement ?

Oui Non

	Oui	Non
L'enfant peut-il valider ? Y-a-t-il un effet ?	X	
La machine "fait-elle la validation" pour l'enfant ?		X
L'enfant constate-t-il la validation de son activité par message du logiciel :		
- Support du message		
. écriture		
. graphisme		
. son		
- Contenu et cas d'envoi du message		
. réponse exacte au problème		
- en cas d'erreur		
- toujours		
. infirmation ou confirmation de la réponse		
. autres éléments		
L'enfant peut-il se sentir maître de la validation ?	X	
. Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?	X	
. Autrement ?		X

Effet de type 2

Y-a-t-il envoi d'une donnée logiciel ?

Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :

- message d'évaluation par rapport à l'attendu conceptuel du logiciel
- donnée d'un score
- donnée de renseignements concernant le temps
- proposition d'un outil spécifique
- proposition de visualisation du problème
- autres instructions

Nombre de types d'effets (1 ou 2)Support réservé à l'évaluation

Graphique écran
Support sonore
Support mobile

Statut des entrées-utilisateur :
degré de programmation laissé à l'utilisateur

Y-a-t-il présence de :

- réponse à la machine
- commande de la machine de type pilotage
- commande de la machine de type programme

L'expression de la réponse au problème traité se fait-elle :

- en mode réponse
- en mode commande

	Oui	Non
Y-a-t-il envoi d'une donnée logiciel ?		X
Dans ce cas, quels sont les objets fournis, du point de vue de l'utilisateur :		
- message d'évaluation par rapport à l'attendu conceptuel du logiciel		
- donnée d'un score		
- donnée de renseignements concernant le temps		
- proposition d'un outil spécifique		
- proposition de visualisation du problème		
- autres instructions		
<u>Nombre de types d'effets (1 ou 2)</u>	1	
<u>Support réservé à l'évaluation</u>	X	
Graphique écran		(X)
Support sonore		(X)
Support mobile		
<u>Statut des entrées-utilisateur :</u> <u>degré de programmation laissé à l'utilisateur</u>		
Y-a-t-il présence de :		
- réponse à la machine		X
- commande de la machine de type pilotage	X	
- commande de la machine de type programme	X	
L'expression de la réponse au problème traité se fait-elle :		
- en mode réponse		X
- en mode commande	X	

ANNEXE IV
Classement de logiciels cités dans la recherche selon la typologie générale

Les logiciels classés sont essentiellement à sémantique procédurale ; toutefois, certains sont à sémantique déclarative.

Le classement est effectué en fonction des traits dominants des logiciels : certains logiciels se distinguent de leur type de référence en ce qui concerne certains critères.

Type A			Type B		Type C	
1 I-I	1 I-II-	1 I-II+	3.1 II-I	3.2 II-I	2 II-	2 II+
CONT 1			CONT 3		CONT 2	
COM I			COM II			
Puce		Rect	Divls	Frise	Pictor	Logo
Angles *		Tache	Simul	Mosaïque	Traitements de textes	Prolog
Jolygones*		Multi	Angles *	Angles*	Tableurs	Basic
Evama			Golf	Jolygones*	Grappeurs	Langages de programmation
Invasion des nombres			Puzzle		Dessin assisté	Générateurs de programmes
Ordre de grandeur			Joligones*			
			Decou			
			Repérages			
			Jeux de cibles			

(* pour un module du logiciel)

ANNEXE V

Principes de schématisation de la structure des logiciels par rapport à la communication

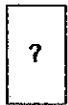
⇒ En faisant référence à la validité des instructions et à la caractérisation des entrées-utilisateur, on établit un schéma suivant les principes évoqués ci-dessous :

- Le logiciel est représenté par un rectangle :



- On signale la présence d'entrées non couvertes par des instructions par un point d'interrogation.

En particulier, si aucune entrée ne peut se faire par instruction, la schématisation est la suivante :



Dans le cas où seules certaines entrées ne peuvent pas se faire par instruction, la schématisation comporte un point d'interrogation encadré :



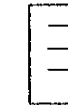
- On signale la présence d'instructions valides sur tout le logiciel, couvrant toutes les entrées possibles, par un coloriage du rectangle représentatif du logiciel :



- On signale la présence d'instructions valides sur une partie du logiciel par un hachurage de rectangles intérieurs au rectangle représentatif du logiciel :



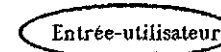
- On signale la présence d'instructions valides à un seul endroit du logiciel par le dessin de segments dans le rectangle représentatif du logiciel :



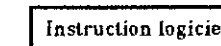
Remarque : les schémas des logiciels comprennent généralement plusieurs des composants qui précèdent.

⇒ En faisant référence à l'organisation, à l'évaluation et au statut des entrées-utilisateur, on établit un schéma suivant les principes évoqués ci-dessous :

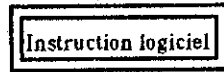
- L'entrée-utilisateur est notée par :



- La présence d'une instruction du logiciel est notée par :



S'il est possible que l'instruction émise par le logiciel soit une évaluation par rapport à un attendu du conceptuel, on effectue un double cadrage :



• La présence d'une production faisant suite et relative à l'entrée utilisateur est notée :



Ceci est caractéristique de logiciels définissant un mode commande.

• Les schémas s'organisent par cycles.

On signale le fait que la réponse s'effectue ou non en mode commande en incluant le terme REPONSE dans un cycle adéquat (cycle avec Production ou Instruction logiciel).

Il en est de même pour la construction éventuelle d'un objet intermédiaire par l'utilisateur : on indique "Objet intermédiaire" dans un cycle approprié.

Voici un exemple de schéma utilisant ces conventions :

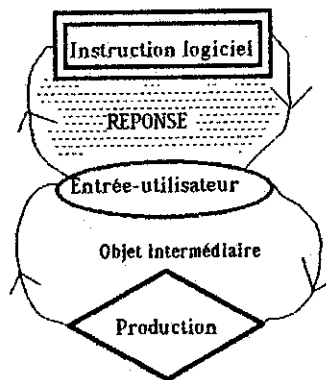


TABLE DES MATIERES



PRESENTATION DE LA RECHERCHE	5
GENERALITES	6
1- Situation de la recherche	6
2- Matériel de la recherche	6
3- Vocabulaire	7
3a Logiciel	7
3b Langage	8
4- Choix des logiciels intervenant dans la recherche	8
OBJECTIFS DE LA RECHERCHE	9
1- Définition des objectifs de la recherche	9
2- Pertinence des objectifs retenus : à propos de la diversité des logiciels	10
2a Les référents terminologiques usuels	10
- Le vocabulaire d'appellation	11
- Le vocabulaire de classification	15
2b Approche usuelle de la diversité des logiciels	19
- Les moyens d'identification des logiciels	20
- Un protocole de choix de logiciels lié aux outils disponibles	21
2c Conclusion	23
HYPOTHESES ET METHODOLOGIE	25
1- Hypothèses	25
2- Méthodologie	25
2a Préalable	25
2b Organisation de la recherche	26
2c Méthodologie spécifique à l'élaboration d'une typologie a priori (Partie I)	26
2d Méthodologie relative à l'expression des enjeux didactiques (Partie II)	27

PARTIE I	
LOGICIELS POUVANT IMPLIQUER DES ACTIVITES	
MATHEMATIQUES A L'ECOLE ELEMENTAIRE :	
TYPOLOGIE A PRIORI	29
TYPOLOGIE CONT	30
1- Préalable	30
2- La grille CONT	31
2a Présentation de la grille CONT	31
2b A propos des rubriques de la grille CONT	34
3- Etude comparative des résultats fournis par l'analyse a priori de logiciels selon la grille CONT	37
3a Etude comparative des résultats des rubriques "Niveau concerné", "Concepts mathématiques impliqués par le logiciel", "Organisation du logiciel"	38
3b Etude comparative des résultats de la rubrique "Action de l'enfant par rapport à l'activité"	41
- Sous-rubrique "Définition du contenu de l'activité"	41
- Sous-rubrique "La donnée des consignes définissant le contenu de l'activité"	44
- Sous-rubrique "La gestion de l'activité"	46
3c Etude comparative des résultats de la sous-rubrique "La validation de l'activité par rapport à son contenu"	48
- Questions concernant les détenteurs de la validation	48
- Questions concernant les bases de la validation	50
4- Définition de la typologie CONT	52
4a Descriptif de la typologie CONT	52
4b Quelques remarques	53
4c Quelques questions	53

TYPLOGIE COM	55
1- Préalable	55
2- La grille COM	55
2a Présentation de la grille COM	55
2b A propos des rubriques de la grille COM	59
3- Etude comparative des résultats fournis par l'analyse a priori de logiciels selon la grille COM	66
3a Etude comparative des résultats de la rubrique "Les instructions pour la réalisation de l'activité : les données du logiciel et de son environnement concernant la communication machine-utilisateur"	67
- Sous-rubrique "Caractérisation"	67
- Sous-rubrique "Support des instructions"	70
- Sous-rubrique "Validité des instructions"	72
3b Etude comparative des résultats de la rubrique "Les entrées-utilisateur"	76
- Sous-rubrique "Caractérisation"	76
- Sous-rubrique "Organisation"	79
- Sous-rubrique "Evaluation"	83
- Sous-rubrique "Statut des entrées-utilisateur : degré de programmation laissé à l'utilisateur"	90
4- Définition de la typologie COM	98
4a Descriptif de la typologie COM	98
4b Quelques remarques	104
4c Quelques questions	106
TYPLOGIE GENERALE CONT/COM	107
1- Liaison des typologies CONT et COM	107
2- Conclusion : définition de la typologie générale	108
2a Préalable	108
2b Descriptif de la typologie générale CONT/COM	108
3- Questions	109

<u>PARTIE II</u>	
<u>LOGICIELS POUVANT IMPLIQUER DES ACTIVITES</u>	
<u>MATHEMATIQUES A L'ECOLE ELEMENTAIRE :</u>	
<u>ENJEUX DIDACTIQUES</u>	111
INTRODUCTION	112
UNE ORGANISATION POSSIBLE DE LA DIVERSITE DES LOGICIELS	114
1- Introduction	114
2- Notre position sur l'usage de moyens informatiques dans l'enseignement	114
3- Apport des grilles d'analyse a priori CONT et COM	115
4- Apport de la typologie générale	116
4a Développement d'une problématique de l'informatique-outil	116
4b Vocabulaire	120
5- Conclusion : évaluation des logiciels	123
LOGICIELS ET COMMUNICATION : COMPLEMENTS	124
1- Rapport entre les instructions de production des logiciels à communication II et les productions qu'elles engendrent	124
1a Logiciels à sémantique déclarative	124
1b Logiciels à sémantique procédurale	125
2- Typologie et instructions de production des logiciels à communication II	129
3- Fonction des instructions de production : conclusion	130

LES TACHES DE COMMUNICATION	133
1- Tâches de communication de type I ; apprentissage	133
1a Position du problème	133
1b Définition des tâches de communication I	133
- Les règles logicielles	135
- Les règles matérielles	136
1c Apprentissage de la communication de type I	137
2- Tâches de communication de type II ; apprentissage	139
2a Position du problème	139
2b Définition des tâches de communication II	140
- Tâche de connaissance des instructions de production	140
- Tâche de connaissance des règles de syntaxe	143
- Tâche de compréhension des messages émis par le logiciel par rapport au discours de l'utilisateur	144
- Tâches méthodologiques	146
2c Apprentissage de la communication de type II	147
- A propos du degré d'apprentissage	148
- A propos des conditions et limites de l'apprentissage	150
- Quelques propositions pour l'apprentissage	153
2d Conclusion relative aux tâches de communication de type II	154
3- Conclusion relative aux tâches de communication	154
A PROPOS DES TACHES RELATIVES AU CONTENU DE L'ACTIVITE	156
1- Participation de l'enfant à la définition de l'activité	156
1a Le principe du "projet" : origine et positions diverses	157
1b Définition de l'activité et principe du "projet"	158
- Existence et illusion du "projet" avec LOGO	159
- Le "projet" et les logiciels à communication II autres que LOGO	161
- Le "projet" comme participation effective de l'élève à la définition de l'activité	162
- Conclusion	163

1c La définition de l'activité par l'élève est-elle une méthode de travail valide ?	164
- Pertinence de la méthode dans la classe, dans la scolarité	164
- Définition de l'activité par l'élève et contenu de formation	164
1d Conclusion : qui doit assurer la définition de l'activité avec un logiciel à communication II	165
2- Statut de l'objet produit avec un logiciel	167
2a Statut de l'objet produit avec un logiciel à communication I	167
2b Statut de l'objet produit avec un logiciel à communication II	168
2c Conclusion	174
3- Validation de l'activité	175
3a Validation de l'activité avec un logiciel à communication I	175
- Messages d'évaluation stricte	176
- Messages autres que d'évaluation stricte	179
- En conclusion	184
3b Validation de l'activité avec un logiciel à communication II	186
- Position du problème	186
- Validation de la composante communication de l'activité	187
- Validation de la composante mathématique de l'activité	187
- En conclusion	195
3c Conclusion	196
4- Conclusion : la gestion des tâches relatives au contenu de l'activité	197

TRANSPOSITION DE CONCEPTS MATHÉMATIQUES ET LOGICIELS	198
1- Types de concepts mathématiques et types de communication	198
1a Type I de communication et concepts mathématiques	199
1b Type II de communication et concepts mathématiques	199
- Domaine géométrique non axiomatisé et domaine numérique	200
- Prospective	201
2- Types de logiciels, de communication, et présentation des concepts mathématiques	202
2a Logiciels de type A (communication de type I)	203
2b Logiciels de type C (communication de type II)	204
2c Logiciels de type B (communication de type II)	207
- Le sous-type 3.2 II-I	207
- Le sous-type 3.1 II-I	208
3- Conclusion	208
SPECIFICITES DES LOGICIELS POUR L'ENSEIGNEMENT DES MATHÉMATIQUES	210
1- Position du problème	210
2- Logiciels pour l'épreuve, la répétition, l'évaluation des savoirs	216
2a Types de logiciels concernés	216
2b Types de situations mises en place	217
2c Types d'intervention dans l'enseignement	218
- Acquisition de savoirs nouveaux par "mécanique objet-outil"	218
- Interrogations sur du savoir ancien	218
- Mise au point	219
2d Domaine de savoirs concernés	219
2e Pertinence des logiciels pour l'épreuve, la répétition, l'évaluation de savoirs	219
- Spécificité : épreuve ou répétition	219
- Spécificité : évaluation	221
2f En guise de conclusion	222

3- Logiciels pour préparer l'institutionnalisation de notions mathématiques	224
3a Types de logiciels concernés	224
3b Les logiciels de type A non-strict pour préparer l'institutionnalisation de notions	224
- Préalable : rapport des élèves au savoir visé	224
- Logiciels A non-strict comme outils didactiques pour préparer l'institutionnalisation de notions	225
- Types de situations et d'intervention dans l'enseignement	226
- Domaine de savoirs concernés	228
3c Les logiciels de types B, C pour préparer l'institutionnalisation de notions	228
- Préalable : rapport des élèves au savoir visé	228
- Types de situations et d'intervention dans l'enseignement	229
- Mise au point	230
- Domaine de savoirs concernés	230
3d En guise de conclusion	231
4- Logiciels pour intervenir sur la gestion de l'activité cognitive	232
4a Position du problème	232
4b Interventions explicites sur la gestion de l'activité	232
- Interventions explicites directes	233
- Interventions explicites indirectes	238
4c Interventions implicites sur la gestion de l'activité	239
4d En guise de conclusion	241
5- Logiciels pour faire respecter des contraintes sur l'expression de la solution au problème posé, ou sur la méthode de sa résolution	242
5a Position du problème	242
5b Logiciels pour faire respecter des contraintes relatives à l'expression de la réponse au problème posé	242
- Logiciels à communication I	243
- Logiciels à communication II	244
5c Logiciels pour faire respecter des contraintes relatives aux procédures à engager pour la résolution du problème posé	247
5d En guise de conclusion	248

6- Logiciels pour accompagner l'enseignement ou l'apprentissage	249
6a Position du problème	249
6b Logiciels pour illustrer	249
6c Logiciels pour manipuler des textes	251
6d Logiciels pour stocker et gérer des informations	252
- Logiciels de gestion	252
- Logiciels permettant l'enregistrement d'actes de l'élève	253
6e En guise de conclusion	253
7- Peut-on définir un nouveau cadre – le cadre logiciel –, pour des concepts mathématiques	254
7a Position du problème	254
7b Cadre logiciel et communication	254
- La communication de type II est susceptible de constituer un cadre	255
- Communication de type II et cadre langage	255
- Communication de type II et cadre descriptif	256
- Sous-types de la communication II, cadre langage, cadre descriptif	257
7c Cadre logiciel et concepts mathématiques	258
7d Les élèves de l'école élémentaire et le cadre langage	259
7e Les élèves de l'école élémentaire et le cadre descriptif	263
- Intellectualisation de l'action	264
- Limites de l'intellectualisation de l'action	265
- A propos de l'objet mathématique d'enseignement	266
7f En guise de conclusion	268

CONCLUSION	269
1- A propos du domaine de validité de la recherche	270
2- A propos de la communication proposée par un logiciel	270
3- Communication et enjeux didactiques des trois types de logiciels	272
3a Préalable concernant les trois types de logiciels	273
3b Les logiciels de type A	273
- Les logiciels de type A strict	273
- Les logiciels de type A non-strict	273
- En conclusion	274
3c Les logiciels de type C	274
3d Les logiciels de type B	276
4- Une prospective de construction de logiciels	277
REFERENCES	279
Références bibliographiques	280
Bibliographie sommaire	283

<u>ANNEXES</u>	285
Liste des annexes	286
Annexe I : descriptif sommaire des neuf logiciels retenus pour l'établissement de la typologie a priori	287
Logiciel PUCE	287
Logiciels SIMUL et RECT	288
Logiciel TACHE	289
Logiciel DIVIS	290
Logiciel MOSAIQUE (MOS)	291
Logiciel FRISE	292
Logiciel PICTOR	293
Logiciel LOGO	293
Annexe II : quelques grilles d'analyse de logiciels	294
Grilles de l'ADI	295
Grille de l'EPI	299
Grille IPT	303
Grille du CESTA	305
Annexe III : analyse a priori selon les grilles CONT et COM des logiciels retenus pour l'établissement de la typologie a priori	335
Logiciel PUCE	336
Logiciel SIMUL	342
Logiciel RECT	348
Logiciel TACHE	354
Logiciel DIVIS	360
Logiciel MOSAIQUE (MOS)	366
Logiciel FRISE	372
Logiciel PICTOR	378
Logiciel LOGO	384
Annexe IV : classement de logiciels cités dans la recherche selon la typologie générale	390
Annexe V : principes de schématisation de la structure des logiciels par rapport à la communication	391
Fascicule d'accompagnement	

LA GRILLE CONT

FASCICULE D'ACCOMPAGNEMENT

La grille CONT s'intéresse aux niveaux d'enseignement, aux types de concepts impliqués par le logiciel, à l'organisation du logiciel par rapport à un concept, à l'action de l'enfant par rapport à l'activité (définition, gestion et validation de l'activité).

Les réponses à apporter sont de type binaire (oui/non), à moins que la question soit sans objet.



Niveau concerné

CP

CE

CM

Concepts mathématiques impliqués par le logiciel

Type arithmétique

Type géométrique

Le logiciel permet-il d'aborder plusieurs concepts de même type ?

Organisation du logiciel par rapport à un concept

Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?

Un nombre fini déterminé ?

Une infinité théorique ?

Les problèmes sont-ils nécessairement isomorphes ?

Oui Non

Action de l'enfant par rapport à l'activité

(sans faire intervenir le problème de communication avec la machine)

La définition du contenu de l'activité

Le logiciel décide-t-il des problèmes à traiter ?

L'enfant ou le maître peuvent-ils à chaque nouveau problème compléter la définition de celui-ci par des consignes faisant intervenir des éléments du logiciel ?

- le complément de définition porte-t-il sur l'objet à produire ?

- le complément de définition porte-t-il sur la démarche à tenir ?

La donnée des consignes définissant le contenu de l'activité

Y-en-a-t-il ?

Sur l'écran ?

Sur une documentation ?

Sont-elles données en une fois pour tout un problème ?

La gestion de l'activité

L'enfant doit-il choisir un type de problème parmi plusieurs qui lui sont proposés ?

L'enfant doit-il choisir un niveau de difficulté parmi plusieurs qui lui sont proposés ?

L'enfant doit-il suivre un cursus de plusieurs problèmes ?

L'enfant doit-il répondre à des questions conceptuelles ?

L'enfant peut-il avoir recours à une aide proposée par le logiciel ?

Oui Non

La validation de l'activité par rapport à son contenu

L'enfant peut-il valider ? Y-a-t-il un effet ?

La machine "fait-elle la validation" pour l'enfant ?

L'enfant constate-t-il la validation de son activité par message du logiciel :

- Support du message
 - . écriture
 - . graphisme
 - . son
- Contenu et cas d'envoi du message
 - . réponse exacte au problème
 - en cas d'erreur
 - toujours
 - . infirmation ou confirmation de la réponse
 - . autres éléments

L'enfant peut-il se sentir maître de la validation ?

- . Par visualisation de la production qu'il a fait faire à la machine en réponse au problème traité ?
- . Autrement ?

COMMENTAIRE DES RUBRIQUES DE LA GRILLE CONT

Il est rappelé que la grille CONT propose une analyse a priori des logiciels du point de vue du contenu, sans émission d'hypothèses relatives aux caractéristiques et enjeux de situations qui emploieraient lesdits logiciels.

La grille CONT n'étudie pas la communication proposée à l'utilisateur avec le micro-ordinateur.

⇒ Rubrique : Niveau concerné

Cette rubrique précise le ou les niveaux de l'école élémentaire auxquels peut s'adresser le logiciel en référence aux concepts qu'il implique.

⇒ Rubrique : Concepts mathématiques impliqués par le logiciel

Cette rubrique s'intéresse au type (arithmétique ou géométrique) des principaux concepts mathématiques mobilisables par le logiciel.

L'implication de concepts peut intervenir :

- soit directement par la donnée de textes de problèmes par le logiciel.
- soit indirectement par la possibilité qu'offre le logiciel de construction d'activités, sans qu'il y ait proposition de textes explicites à l'écran.

⇒ Rubrique : Organisation du logiciel par rapport à un concept

Il s'agit d'examiner du point de vue du nombre et de la dépendance les problèmes que le logiciel pose ou peut permettre de poser.

"Isomorphe" signifie ici, données différentes, problème inchangé.

⇒ Rubrique : Action de l'enfant par rapport à l'activité

Sous-rubrique : Définition du contenu de l'activité

On dira qu'un des protagonistes définit le contenu de l'activité, si c'est lui qui décide du problème à traiter.

- Le logiciel peut le faire complètement ou partiellement, c'est-à-dire

qu'il sera alors possible au maître ou à l'enfant de compléter la décision logicielle.

Notons que la question de la définition du contenu de l'activité est à examiner séparément de celle de la formulation des consignes : en effet, la définition par le logiciel peut-être explicite (avec émission de textes) ou implicite (sans émission de textes).

– Le logiciel peut aussi être tel que son contenu fasse référence à un ou plusieurs concepts sans pour autant qu'il décide des problèmes à traiter : l'enfant ou le maître peuvent alors décider et formuler l'activité à des degrés divers de prise en charge, suivant le logiciel.

Toutes ces questions concernent les possibilités de formulation de l'activité par le maître et la classe, avec des moyens oraux ou graphiques. On ne s'intéresse pas ici à des interventions sur le logiciel même.

Sous-rubrique : La donnée des consignes définissant le contenu de l'activité

On examine les éventuels supports de consignes attachés au logiciel.

La liberté d'émission de consignes orales, ou graphiques se déduit des réponses apportées à cette rubrique et à la précédente.

Sous-rubrique : La gestion de l'activité

Cette rubrique ne donne lieu à réponses que dans le cas où le logiciel définit les problèmes à traiter.

Signalons aussi que la rubrique suivante étudie à part la validation de l'activité.

On dira que "l'enfant doit répondre à des questions conceptuelles" si le logiciel propose des questions relatives à l'attendu conceptuel, qui soient intermédiaires entre la donnée du problème et sa solution.

On dira que "l'enfant peut avoir recours à une aide proposée par le logiciel", s'il existe une partie spécifique du logiciel, distincte du logiciel entier, à vocation d'aide pour la donnée de la réponse au problème traité. Pour préciser, cette partie est (ou pourrait être) optionnelle.

A titre d'exemple, l'aide pourrait consister en :

- la possibilité de modifier, de gérer l'écriture du problème.
- la possibilité de visualiser le problème, d'en avoir une autre expression.
- la possibilité de réaliser une action annexe propre à aider à la résolution du problème.

On rappelle que l'on ne juge pas ici de la pertinence de cette aide.

Sous-rubrique : Validation de l'activité par rapport à son contenu

On ne cherche pas à distinguer évaluation de validation, ni à cerner de façon précise le processus de validation.

Les questions ont seulement pour but de préciser dans quelle mesure l'enfant peut, ne serait-ce que se poser le problème de la validation de son activité.

Par conséquent, on examine tout d'abord s'il y a un effet suite au traitement qu'a effectué l'enfant du problème posé : il y a un effet s'il se passe quelque chose ayant un rapport avec le traitement.

Remarquons que si le logiciel permet une intervention du maître ou de l'enfant dans la définition des problèmes à traiter, on peut avoir affaire à des consignes au niveau de l'objet à produire, mais aussi au niveau de conditions à respecter pour l'obtenir ; dans ce cas, il peut y avoir effet au niveau de l'objet, mais éventuellement non au niveau de la démarche : pour ces logiciels, on se contentera de répondre à la question au niveau de l'objet à produire (ce qui peut être le contenu complet de l'activité). On restera conscient de la lacune éventuelle au niveau de la validation du respect des conditions : ce point sera repris par la formulation "se sentir maître de la validation".

Si le logiciel ne permet pas une intervention extérieure dans la définition des problèmes à traiter, la question de l'effet qui permet de valider est posée pour le contenu complet de l'activité tel qu'il est défini par le logiciel.

En cas de réponse positive à la question concernant l'effet, on recherche par des questions qui ne sont pas exclusives sur qui repose la validation.

Par rapport à l'effet qui permet de valider, le problème est : qui produit cet effet ?

– La machine ?

Elle peut le faire sans pour autant donner de réponse au problème traité (infirmité ou confirmation de la réponse seulement).

La machine peut aussi apporter des éléments en fournissant une autre expression de la réponse de l'enfant.

Il peut exister d'autres données attachées au message de validation (en particulier, indications relatives au temps, envoi de scores, ...) : elles pondèrent la validation, mais ne sont pas la validation à strictement parler. Leur présence est signalée par "autres éléments" et sera reprise par la grille COM.

– L'enfant ?

Expliquons la formulation "se sentir maître de la validation" :

L'élève peut penser valider, et en fait n'être pas en mesure de le faire réellement soit au niveau de l'objet, – c'est ainsi que des enfants de CE1 ont

validé comme carré un graphisme produit sous LOGO par une série d'instructions où les rotations étaient de 89° -, soit au niveau des contraintes de production de l'objet.

Il peut aussi être tenté de valider une réponse à un autre problème que celui initialement posé (production d'un objet intermédiaire par exemple).

La visualisation de la production (écriture, graphisme) semble être le point d'ancrage de la validation par l'enfant ; la rubrique "autrement" est ouverte pour tenir compte d'éventuels autres moyens (sonores, mobiles, ...).

DESCRIPTIF DE LA TYPOLOGIE CONT

L'étude des résultats fournis par l'analyse a priori des neuf logiciels selon la grille CONT permet de proposer une typologie en trois types.

Celle-ci présente deux pôles (le type 1 et le type 2) et un type intermédiaire (le type 3).

Les trois types de la typologie CONT se caractérisent ainsi :

Le type 1

Logiciels impliquant un concept arithmétique ou géométrique concernant un nombre fini de problèmes isomorphes,

- définissant complètement le contenu de l'activité et l'exprimant à l'écran par des consignes données en une fois ou au fur et à mesure,
- intervenant dans la gestion de l'activité par des propositions directes,
- effectuant la validation de l'activité.

Le type 2

Logiciels impliquant une infinité de problèmes non isomorphes concernant plusieurs concepts arithmétiques ou géométriques,

- laissant à autrui la définition complète du contenu de l'activité et son expression,
- n'intervenant pas dans la gestion de l'activité par des propositions directes,
- permettant à l'enfant de se sentir maître de la validation.

Le type 3

Ce type comporte deux sous-types : les sous-types 3.1 et 3.2.

Caractéristiques générales du type 3

Logiciels définissant partiellement le contenu de l'activité,

- l'exprimant à l'écran par des consignes ou laissant autrui l'exprimer,
- permettant à l'enfant de se sentir maître de la validation.

Caractéristiques spécifiques au sous-type 3.1

Logiciels impliquant un nombre fini de problèmes isomorphes sur un concept arithmétique ou géométrique.

Le complément de définition est possible sur la démarche.

Le logiciel intervient dans la gestion de l'activité par des propositions directes.

Caractéristiques spécifiques au sous-type 3.2

Logiciels impliquant une infinité de problèmes non isomorphes concernant plusieurs concepts géométriques.

Le complément de définition est possible sur l'objet de l'activité (et a fortiori sur la démarche).

Le logiciel n'intervient pas dans la gestion de l'activité par des propositions directes.

LA GRILLE COM

La grille COM recourt à la notion d'«instruction» que nous avons jugé propre à l'étude de la communication de l'utilisateur avec le micro-ordinateur. Elle s'intéresse également à la nature, l'organisation, l'évaluation, le statut des entrées de l'utilisateur.

Les réponses à apporter sont de type binaire (oui/non), à moins que la question soit sans objet.

**Les instructions pour la réalisation de l'activité :
les données du logiciel et de son environnement
concernant la communication machine-utilisateur**

Oui Non

Caractérisation

Y-a-t-il :

- a - instructions de définition de l'activité ?
- b - instructions questions à l'utilisateur par rapport au problème qu'il traite ?
- c - instructions de parcours du logiciel ?
- d - instructions qui permettent à l'utilisateur de faire produire la machine ?

Support des instructions

Y-a-t-il des instructions (autres qu'une invite utilisateur) :

Sur l'écran ?

Le périphérique d'entrée rend-il ce support nécessaire ?

Sur un document ?

Sur un autre type de support ?

Validité des instructions

Pour les instructions qui peuvent être des entrées utilisateur (instructions de type d et c - branchement dans le logiciel)

Toutes sont-elles valides sur tout le logiciel ?

Certaines sont-elles valides sur une partie du logiciel ?

Certaines sont-elles valides à un seul endroit du logiciel ?

Les entrées-utilisateur

Oui Non

Caractérisation

Une entrée-utilisateur peut-elle être :

- une instruction ?
- une combinaison d'instructions ?
- autre chose qu'une instruction ou une combinaison d'instructions ?

Organisation

Peut-on enchaîner des entrées :

- sans recevoir d'instructions autres qu'un symbole d'invite ?
- sans recevoir d'instructions autres que la précédente instruction ?
- sans provoquer d'évaluation ?

Evaluation

Caractérisation

Y-a-t-il une évaluation syntaxique/sémantique :

- par rapport au discours-utilisateur ?
- par rapport à l'attendu du conceptuel du logiciel ?

Effet de l'évaluation

Etude de la production de la machine suite à une entrée utilisateur valide (c'est-à-dire ne donnant pas lieu à un message de correction syntaxique/sémantique par rapport au discours-utilisateur) :

Effet de type 1

Y-a-t-il exécution avec production d'objets commandés par l'entrée-utilisateur ?

La sous-rubrique "caractérisation" invite à classer les instructions qu'elle examine en quatre types (a - b - c - d).

Précisons ces quatre types :

Instructions que l'utilisateur peut recevoir de la machine	Instructions que l'utilisateur peut donner à la machine
type a : instructions de définition de l'activité	type d : instructions de production
type b : instructions questions à l'utilisateur par rapport au problème qu'il traite	type c : instructions de parcours du logiciel - type branchement dans le logiciel
type c : instructions de parcours du logiciel - type repérage dans le logiciel	

• Les instructions de type c (instructions de parcours du logiciel) sont de deux sortes :

- type repérage dans le logiciel : elles servent à préciser les entrées que l'utilisateur pourra employer par la suite.
- type branchement dans le logiciel : sans implication réelle dans une production ultérieure, elles ne font que donner accès à certaines parties du logiciel. Elles constituent l'expression d'un choix (de niveau de difficulté, par exemple) et ne sont pas significatives pour la suite de l'activité.

• Les instructions de type d (instructions de production) sont constituées :

- des instructions qui permettent à l'utilisateur de faire produire immédiatement la machine.
- des instructions qui sont en fait des instructions de branchement, mais qui ont une implication directe dans une production : elles sont significatives dans celle-ci.

QUELQUES REMARQUES, EXEMPLES, CONTRE-EXEMPLES

Certaines instructions se présentent sous forme de phrases.

Voici deux exemples :

- "Tu vas faire des divisions" (instruction de type a).

- "Choisis un niveau (A-B)" (instruction de type c - repérage dans le logiciel).

D'autres instructions se présentent sous forme de codages.

Toutefois, tous les codages ne sont pas des instructions au sens défini pour cette recherche.

Examinons les codages par lettres, nombres et expression "oui/non".

Il convient de distinguer :

- les codages qui traduisent des données brutes du logiciel.

Ils seront retenus comme instructions.

On trouve :

- des codages locaux, sans rapport explicite entre signifiant et signifié concernés : ainsi, le mot "oui" ou le chiffre 1 peuvent coder des faits différents sur l'ensemble du logiciel.

On n'oubliera pas qu'ils s'utilisent généralement en réponse à une demande de la machine.

Ils concernent essentiellement des choix de situation.

Exemples :

- Choisis un niveau (A-B)
- Veux-tu faire des additions ? (oui-non)
- "A-B-oui-non" sont ici des instructions de type c - branchement dans le logiciel.

- des codages valides pendant la totalité du logiciel. Un code traduit une donnée brute du logiciel et une seule.

Il y a souvent un rapport explicite entre signifiant et signifié (initiale par exemple). Ils sont généralement réalisés par lettres et ne s'utilisent pas nécessairement en réponse à une demande de la machine.

Exemples :

- Pour donner ta réponse, tape R
- Pour que la machine dessine une ligne, tape L
- "R" est une instruction de type c - branchement dans le logiciel ;
- "L" est une instruction de type d.

Remarquons qu'il peut exister des codages de données brutes valides sur tout le logiciel, réalisés par autre chose qu'une lettre (graphisme sur écran - cas des instructions LOGO par exemple).

- les codages qui traduisent des constructions personnelles de l'utilisateur indépendamment des moyens logiciels.

Ils ne seront pas retenus comme instructions.

Exemples :

- Le nombre 21 est-il :
 - A - multiple de 7 ?
 - B - multiple de 5 ?
 Donne ta réponse (A ou B).
 - Pour résoudre ce problème, faut-il faire une addition ? (oui - non)
- "A-B-oui-non" ne sont pas dans ces cas des instructions.
D'une façon générale, les codages de réponses des logiciels QCM ne sont pas des instructions au sens défini pour cette recherche.

CAS PARTICULIERS

- On appellera «invite-utilisateur» le signal qui fait que l'utilisateur sait qu'il peut effectuer une entrée.

Il s'agit généralement d'un symbole ou du curseur, éventuellement de l'attente de la machine suite à ce qui précède (invite implicite).

L'instruction qui précède le cas échéant l'invite à proprement parler n'est pas incluse dans l'invite.

D'après le sens donné au mot instruction, l'invite constitue une instruction. Attendu qu'elle devrait s'exprimer de la même façon tout au long du logiciel, elle ne sera pas prise en considération.

- Une instruction produite par la machine demandant le nom de l'utilisateur (ou d'autres renseignements sur son compte) ne sera pas retenue lors de l'analyse d'un logiciel : bien que significative pour les rapports machine-utilisateur, elle n'entre pas dans l'activité proprement dite. L'entrée-utilisateur qui lui correspond ne sera pas non plus étudiée.

Sous-rubrique : Support des instructions

Ces supports peuvent être variés. Certains périphériques d'entrée (crayon-optique, souris ...) peuvent rendre le support écran nécessaire.

Sous-rubrique : Validité des instructions

Les instructions de types a, b et c-repérage sont données par le logiciel, et généralement valides une fois, ou au mieux périodiquement sur le logiciel.

Les instructions de types d et c-branchement peuvent être données par l'utilisateur à la machine (elles peuvent être des entrées-utilisateur) et l'on constate qu'elles peuvent avoir un plus grand domaine de validité : la sous-rubrique "validité des instructions" étudie la validité des instructions qui peuvent être des entrées-utilisateur.

De façon plus précise, on distingue trois domaines de validité :

- toutes les instructions sont valides sur tout le logiciel.

- Certaines instructions sont valides sur une partie du logiciel.
La partie en question peut être constituée d'une réunion de diverses parties disjointes.

- Certaines instructions sont valides à un seul endroit du logiciel.

Ainsi, l'examen de la validité des instructions renseigne sur la possibilité qu'a l'utilisateur d'effectuer une entrée avec des instructions diverses.

Lorsqu'un même problème se répète plusieurs fois, il ne faut répondre aux questions de cette sous-rubrique que pour un exemplaire du problème.

⇒ Rubrique : Les entrées-utilisateur

Dans cette partie, le terme écriture s'entend au sens large, c'est-à-dire qu'il s'agit d'une écriture réalisée avec un périphérique d'entrée de la configuration (clavier, périphérique vocal, analogique, stylo-optique, etc ...).

Sous-rubrique : Caractérisation

On caractérise les entrées-utilisateur en fonction de la notion d'instruction telle qu'elle a été définie précédemment.

Pratiquement, une entrée peut être :

- une instruction de type d (instruction de production) ou c-branchement dans le logiciel.

- une «réponse», ce qui est désigné par "autre chose qu'une instruction ou une combinaison d'instructions" ; dans ce cas, il peut y avoir nécessité de construction personnelle de l'utilisateur, indépendamment des moyens logiciels.

- une combinaison d'instructions (examen d'une amorce de programmation).

Sous-rubrique : Organisation

On s'intéresse à la possibilité pour l'utilisateur d'enchaîner des entrées, et on tente d'apprécier par trois questions le degré d'intervention explicite du logiciel dans le discours-utilisateur.

Les deux premières questions cherchent à préciser s'il y a émission par le logiciel d'instructions suite à une entrée-utilisateur. Remarquons que lesdites instructions font partie de ce que nous appelons l'évaluation de l'entrée-utilisateur. Néanmoins, ici, il s'agit simplement de repérer la présence ou l'absence d'instructions données par le logiciel suite à une entrée de l'utilisateur, le problème général de l'évaluation étant traité par la

sous-rubrique suivante.

La dernière question de cette sous-rubrique concerne l'existence d'un mode différé.

Remarque :

L'examen des réponses aux rubriques "caractérisation des entrées", "organisation des entrées" et "validité des instructions" permet d'évaluer si l'utilisateur est maître des entrées et de leur ordre (dans le cadre d'un respect des règles).

Sous-rubrique : Evaluation

On se place du point de vue de l'utilisateur et on veut préciser ce qu'est l'évaluation de ses entrées.

Dans le cas où les entrées auraient été faites en mode différé, l'étude se situe à partir de l'exécution.

Le paragraphe "Effet de type 1" repère la production d'objets comme évaluation d'une entrée-utilisateur.

Le paragraphe "Effet de type 2" fait état de diverses données logicielles (c'est-à-dire instructions au sens défini pour cette recherche), qui peuvent apparaître.

Suite à la définition des divers types d'instructions, on peut dire que lorsqu'une entrée est une instruction, elle donne lieu à un effet de type 1 si elle est de type d (instruction de production), et à un effet de type 2 si elle est de type c (instruction de parcours-type branchement dans le logiciel).

Sous-rubrique : Statut des entrées-utilisateur ; degré de programmation laissé à l'utilisateur

Cette partie constitue une synthèse de questions précédentes et étudie la façon dont s'exprime la réponse au problème traité.

On dira qu'il y a présence de :

– "réponse à la machine", s'il existe des entrées-utilisateur qui ne sont pas des instructions (voir la rubrique de caractérisation des entrées).

– "commande de la machine de type pilotage", s'il est possible d'effectuer des entrées avec des instructions de type d (voir les rubriques de caractérisation des instructions et des entrées).

– "commande de la machine de type programme", s'il est possible d'enchaîner des entrées faites avec des instructions de type d (instructions de production) sans provoquer d'évaluation (voir les rubriques de caractérisation des instructions et des entrées).

L'analyse d'un logiciel selon cette sous-rubrique de la grille COM permet de préciser dans quelle mesure il est possible à un utilisateur d'exprimer un algorithme qu'il a construit.

Remarque :

Certains logiciels autorisent l'emploi d'instructions de parcours (type c) pour réaliser des entrées.

Ce genre d'entrées n'est pas retenu pour définir le mode commande.

En effet, serait alors conçu un mode commande qui porterait sur le parcours du logiciel. Il nous a semblé préférable de réserver le terme de «mode commande» pour ce qui concerne le contenu de l'activité. Le mode d'expression du parcours du logiciel, – qu'il serait d'ailleurs possible de modifier –, accentuera le cas échéant le caractère spécifique du logiciel. C'est en ce sens que l'on en tiendra compte.

PRINCIPES DE SCHEMATISATION DE LA STRUCTURE DES LOGICIELS PAR RAPPORT A LA COMMUNICATION

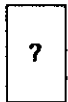
⇒ En faisant référence à la validité des instructions et à la caractérisation des entrées-utilisateur, on établit un schéma suivant les principes évoqués ci-dessous :

- Le logiciel est représenté par un rectangle :



- On signale la présence d'entrées non couvertes par des instructions par un point d'interrogation.

En particulier, si aucune entrée ne peut se faire par instruction, la schématisation est la suivante :



Dans le cas où seules certaines entrées ne peuvent pas se faire par instruction, la schématisation comporte un point d'interrogation encadré :



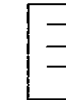
- On signale la présence d'instructions valides sur tout le logiciel, couvrant toutes les entrées possibles, par un coloriage du rectangle représentatif du logiciel :



- On signale la présence d'instructions valides sur une partie du logiciel par un hachurage de rectangles intérieurs au rectangle représentatif du logiciel :



- On signale la présence d'instructions valides à un seul endroit du logiciel par le dessin de segments dans le rectangle représentatif du logiciel :



Remarque : les schémas des logiciels comprennent généralement plusieurs des composants qui précèdent.

⇒ En faisant référence à l'organisation, à l'évaluation et au statut des entrées-utilisateur, on établit un schéma suivant les principes évoqués ci-dessous :

- L'entrée-utilisateur est notée par :

Entrée-utilisateur

- La présence d'une instruction du logiciel est notée par :

Instruction logiciel

S'il est possible que l'instruction émise par le logiciel soit une évaluation par rapport à un attendu du conceptuel, on effectue un double cadrage :

Instruction logiciel

• La présence d'une production faisant suite et relative à l'entrée utilisateur est notée :



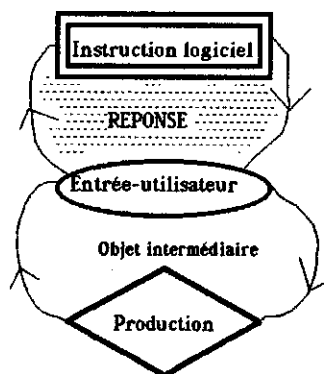
Ceci est caractéristique de logiciels définissant un mode commande.

• Les schémas s'organisent par cycles.

On signale le fait que la réponse s'effectue ou non en mode commande en incluant le terme REPONSE dans un cycle adéquat (cycle avec Production ou Instruction logiciel).

Il en est de même pour la construction éventuelle d'un objet intermédiaire par l'utilisateur : on indique "Objet intermédiaire" dans un cycle approprié.

Voici un exemple de schéma utilisant ces conventions :



DESCRIPTIF DE LA TYPOLOGIE COM

L'étude des résultats fournis par l'analyse a priori des neuf logiciels selon la grille COM permet d'établir une typologie.

Celle-ci comporte deux types : les types I et II.

Chaque type comporte des sous-types.

Les sous-types I-I du type I et II⁺ du type II sont des pôles de la typologie COM.

La description de chaque sous-type comporte une schématisation relative à la structure des logiciels. Le principe de cette schématisation est donné en annexe V et dans le fascicule d'accompagnement.

Le type I

Ce type comporte deux sous-types (I-I et I-II), le sous-type I-II se décomposant lui-même en deux sous-types (I-II⁻ et I-II⁺).

Caractéristiques générales du type I

Les logiciels de ce type sont des logiciels «réponse».

La réponse au problème traité s'effectue en mode réponse.

L'utilisateur ne dispose pas d'instructions qui lui permettent d'exprimer un algorithme relativement à la réponse, d'anticiper, de programmer.

Il y a évaluation par rapport à l'attendu conceptuel.

Caractéristiques spécifiques au sous-type I-I

Avant l'emploi, l'utilisateur ne dispose de rien.

Les instructions sont données par le logiciel à l'écran.

Elles peuvent être de plusieurs genres, à l'exclusion du genre instruction de production.

En règle générale, les entrées utilisateurs ne sont pas des instructions. Elles ne peuvent l'être qu'exceptionnellement et concernent alors exclusivement le parcours du logiciel.

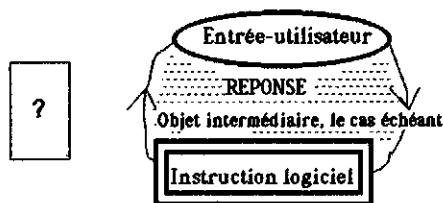
L'utilisateur est toujours en mode réponse, qu'il s'agisse de produire la solution du problème ou un objet intermédiaire. Il n'y a souvent qu'une seule possibilité par entrée-utilisateur, exigeant une construction personnelle de l'utilisateur (n'employant pas de données brutes du logiciel ou de son environnement).

Suite à une entrée, l'utilisateur reçoit une instruction.

Dans ce cadre, l'utilisateur ne décide rien.

Il n'y a aucune possibilité d'expression d'algorithme, d'anticipation, de programmation.

Schématisation



Caractéristiques spécifiques au sous-type I-II

Avant l'emploi, l'utilisateur peut disposer de renseignements.

Les instructions sont données à l'écran ou sur un document. Elles peuvent être de plusieurs genres : le genre parcouru est toujours présent ; le genre production également, mais il concerne la construction d'un objet intermédiaire.

Si celui-ci est toujours optionnel, on parlera du **sous-type I-II⁻**.

Si celui-ci est obligatoire sur au moins une partie du logiciel, on parlera du **sous-type I-II⁺**.

Il existe au moins un groupe d'instructions valides sur une partie du logiciel, ou des groupes d'instructions valides chacun à des endroits différents du logiciel.

Certaines entrées-utilisateur sont des instructions.

L'utilisateur est en mode commande de type pilotage pour la réalisation d'un objet intermédiaire et en mode réponse pour la réponse au problème traité.

Suite à une entrée, il y a production et/ou instruction.

Dans ce cadre, localement, l'utilisateur peut disposer d'éléments et décider.

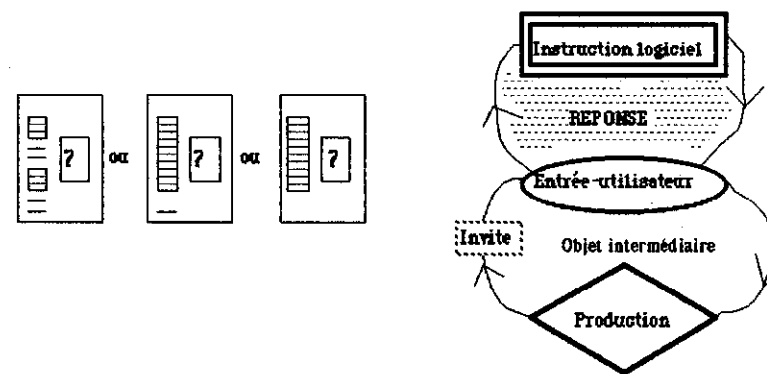
Il y a possibilité d'exprimer un algorithme, d'anticiper, de programmer pour la réalisation d'un objet intermédiaire. Néanmoins, l'anticipation n'est pas une donnée directement attachable au logiciel : il n'y a pas de mode différé.

Schématisations

On peut réaliser deux sortes de schémas pour le sous-type I-II.

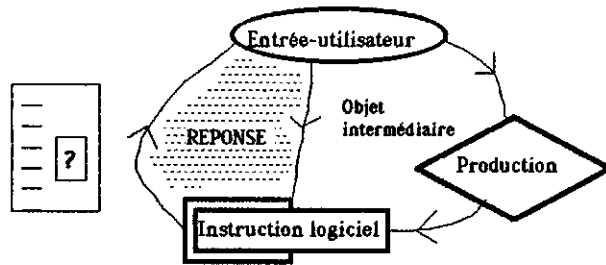
Notons qu'en sous-type I-II⁺, les deux genres de cycles sont nécessairement employés avec dominance des cycles production. En sous-type I-II⁻, les deux genres de cycle ne sont pas nécessairement employés ; il y a dominance des cycles réponse.

– Schéma a



Remarque : l'instruction invite, si elle est présente, est la même sur plusieurs cycles.

– Schéma b



Le type II

Ce type comporte trois sous-types (II-I, II⁻ et II⁺).

Caractéristiques générales du type II

Les logiciels de ce type sont des logiciels «commande».

La réponse au problème traité s'effectue en mode commande. Avant l'emploi, l'utilisateur peut disposer de renseignements. Il n'y a pas d'évaluation par rapport à l'attendu du conceptuel.

Dans le cadre précisé par chaque sous-type, globalement ou plus ou moins localement, l'utilisateur dispose d'éléments et décide.

Il y a une possibilité d'exprimer un algorithme, d'anticiper, de programmer pour la réponse au problème traité ou la réalisation d'un objet intermédiaire.

Caractéristiques spécifiques aux sous-types II⁻ et II⁺

Les instructions sont données par un document.

Elles sont d'un seul genre : instructions de production.

Elles sont valides sur tout le logiciel.

Les entrées utilisateur sont des instructions.

L'utilisateur dispose toujours d'un mode commande de type pilotage.

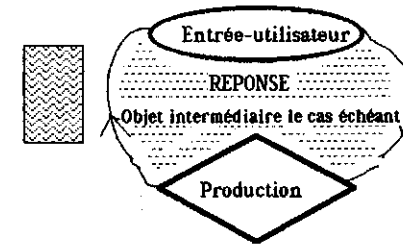
Suite à une entrée, l'utilisateur ne reçoit pas d'instructions.

• Caractéristiques spécifiques au sous-type II⁻

Suite à une entrée, il y a toujours production.

L'anticipation n'est pas une donnée directement attachable au logiciel : il n'y a pas de mode différé.

Schématisation



• Caractéristiques spécifiques au sous-type II⁺

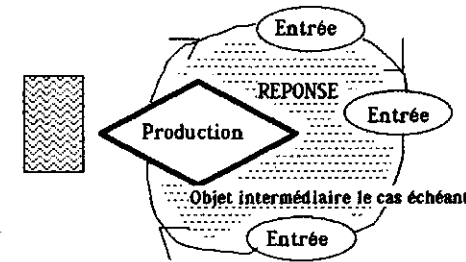
Il existe un mode différé.

L'anticipation est une donnée directement attachable au logiciel.

Schématisation

La schématisation du sous-type II⁺ relative au mode pilotage est celle du sous-type II⁻.

Schématisation relative au mode différé



• Caractéristiques spécifiques au sous-type II-I

Les instructions sont données à l'écran ou sur un document.

Elles peuvent être de plusieurs genres : les genres parcours et production sont toujours présents.

Il existe au moins un groupe d'instructions valides sur une partie du logiciel ou des groupes d'instructions valides chacun à des endroits différents du logiciel.

En règle générale, les entrées utilisateur sont des instructions.

Elles ne peuvent être qu'exceptionnellement des réponses et concernent

alors exclusivement le parcours du logiciel.

L'utilisateur est toujours en mode commande de type pilotage pour la réponse au problème traité ou la réalisation d'un objet intermédiaire.

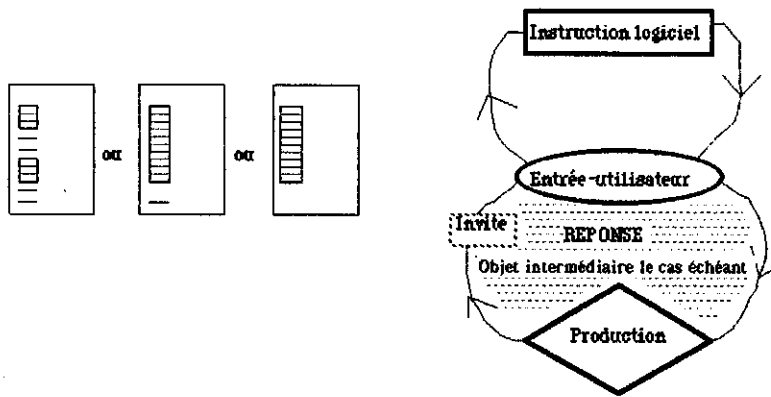
Suite à une entrée, il y a production et/ou instruction.

L'anticipation n'est pas une donnée directement attachable au logiciel : il n'y a pas de mode différé.

Schématisations

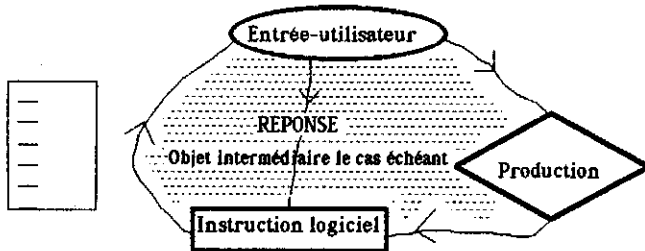
On peut réaliser deux sortes de schémas.

- Schéma a

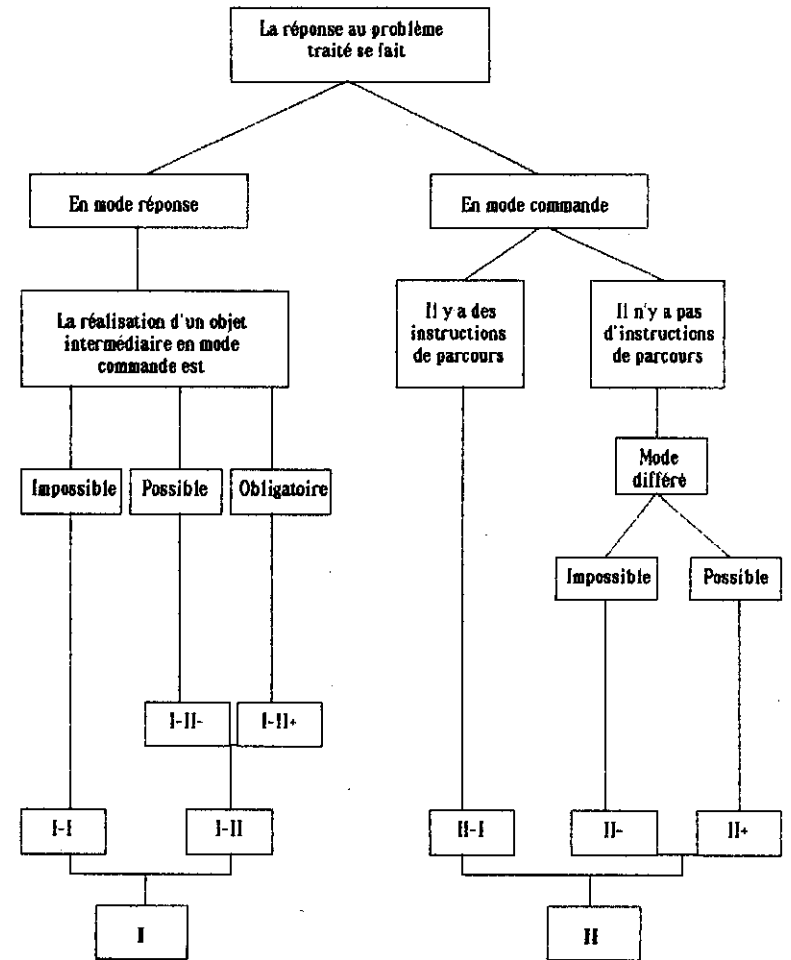


Remarque : l'instruction invite, si elle est présente, est la même sur plusieurs cycles.

- Schéma b



ARBRE DE PREMIER CLASSEMENT SELON LES TYPES DE LA TYPOLOGIE COM



TYPOLOGIE GENERALE CONT/COM

Il est possible de proposer une typologie générale en trois types avec sous-types.

Pour retrouver les caractéristiques de chaque type, il convient de se reporter à l'exposé des typologies CONT et COM qui figurent dans ce fascicule : les chiffres en écriture arabe renvoient aux types de CONT, ceux en écriture romaine aux types de COM.

Le type A

- . sous-type 1 I-I
- . sous-type 1 I-II⁻
- . sous-type 1 I-II⁺

Le type B

- . sous-type 3.1 II-I
- . sous-type 3.2 II-I

Le type C

- . sous-type 2 II⁻
- . sous-type 2 II⁺

REMARQUES

On peut avoir le même type pour le contenu et des sous-types différents pour la communication (en A et C).

On peut avoir le même type pour la communication et de sous-types différents pour le contenu (en B).

Toutefois, au niveau des types mêmes, il semble qu'il y ait corrélation entre contenu et communication :

- Le type 1 de CONT est attaché au type I de COM.
- Les types 2 et 3 de CONT sont attachés au type II de COM ; plus

précisément, le type 2 de CONT est attaché aux sous-types II de COM sans caractéristiques I, tandis que le type 3 de CONT l'est au sous-type de COM avec caractéristiques I.

Les deux tableaux ci-dessous ont permis, et rendent compte de ces remarques :

	Type A	Type B	Type C
CONT	1	3	2
COM	I	II II-I II- II ⁺	

	Type A	Type B	Type C
CONT	1	3.1 3.2	2
COM	I-I I-II- I-II ⁺	II-I	II- II ⁺

VOCABULAIRE D'USAGE ET TYPOLOGIE

TERMINOLOGIE

D'après le vocabulaire d'usage		D'après la typologie générale	
EPO	Didacticiels OCM	Type A 1 I-I	Logiciels réponse Pas de langage Mini langage de commande disponible pour objet intermédiaire
	Didacticiels	1 I-II-	
	Didacticiels	1 I-II+	
E	Didacticiels Imagiciels	Type B 3.1 II-I	Logiciels commande Langage de commande
	Imagiciels Logiciels-outils	3.2 II-I	
	Logiciels-outils Progiciels Logiciels intégrés Tableaux Traitements de textes Gestionnaires de données	Type C 2 II-	
A O	Langage auteur Langages de programmation Générateurs de programmes	2 II+	Langage de programmation

HUIT SYSTEMES D'INSTRUCTIONS DE PRODUCTION DE LOGICIELS A SEMANTIQUE PROCEDURALE

Les instructions qui assurent la production d'un objet O_i impliqué par un objet O correspondent à :

□ l'objet O_i . $S_1-[O_i]$

Ainsi avec le logiciel LOGO, complété d'une procédure prédéfinie RECTANGLE, l'objet O_i "rectangle" peut s'obtenir par l'instruction RECTANGLE, c'est-à-dire directement l'instruction "objet O_i ".

L'instruction "objet O_i " signifie construire l'objet O_i désigné par l'instruction même (et en conséquence désigné par l'utilisateur), avec des paramètres définis par le logiciel. Elle comprend donc implicitement tout ce qui est nécessaire à la construction de l'objet, ceci étant totalement en charge du logiciel. La réalisation est transparente pour l'utilisateur; il n'a pas à décrire les tâches nécessaires à la production. Les paramètres de l'objet peuvent néanmoins lui apparaître, suite à son affichage.

□ un ou plusieurs paramètres p_i de l'objet O_i . $S_2-[p_i]$

Ainsi avec le logiciel JOLIGONES, pour obtenir un polygone convexe régulier, on peut donner comme instructions deux de ses paramètres, par exemple le nombre de ses côtés et la mesure commune de ses angles.

Les instructions "paramètre p_i " signifient construire avec ces paramètres désignés par l'utilisateur un objet O_i défini par le logiciel.

□ l'objet O_i et certains de ses paramètres. $S_3-[O_i, p_i]$

Ainsi avec le logiciel PICTOR, pour obtenir un rectangle aux côtés parallèles aux bords de l'écran, on peut fournir comme instructions "objet rectangle" que l'on désigne sur la barre du menu, et deux de ses paramètres que sont deux sommets opposés que l'on pointe sur l'écran.

Avec le logiciel LOGO et une procédure à paramètres (longueur,

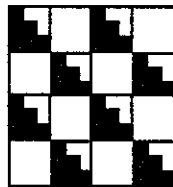
largeur) pour la réalisation d'un rectangle, l'instruction RECTANGLE 3 12 revient pour l'utilisateur à désigner l'objet et deux de ses paramètres. On pourrait aussi citer dans ce cadre les logiciels permettant la représentation graphique de fonctions lorsqu'on peut en désigner une parmi plusieurs; dans le cas contraire, le logiciel propose un système de type S2.

L'explicitation du système S3 est identique à celle du système S2, avec cette différence: l'objet O_i est ici désigné par l'utilisateur.

Remarque: Les systèmes S1, S2, S3 recouvrent toutes les possibilités de combinaison à partir des données O_i , p_i .

un ou des sous-objets O_{bi} de O_i en même temps que des opérations implicites O_{pi} sur ceux-ci. **S4-[O_{bi} , O_{pi}]**

Ainsi avec le logiciel MOSAIQUE, pour produire l'objet O_i ci-dessous, on peut donner comme instructions "sous-objet motif" (avec des paramètres que sont couleur du fond et du motif), ce qui inclut des opérations de translation sur ce dernier.

Objet O_i 

Sous-objet

L'instruction "sous-objet O_{bi} " signifie, construire avec le sous-objet désigné et des opérations implicites un objet O_i défini par le logiciel.

Elle comprend donc implicitement la donnée d'opérations définies par le logiciel et attachées au sous-objet. Celles-ci peuvent apparaître à l'utilisateur lors de l'affichage, leur réalisation lui restant transparente.

des opérations O_{pi} en même temps que des sous-objets implicites O_{bi} sur lesquels elles s'appliquent. **S5-[O_{bi} , O_{pi}]**

Ainsi avec le logiciel DIVIS, on peut donner l'instruction "opération -5" qui s'appliquera à un stade donné sur le sous-objet constitué du nombre résultant des soustractions successives effectuées jusqu'alors.

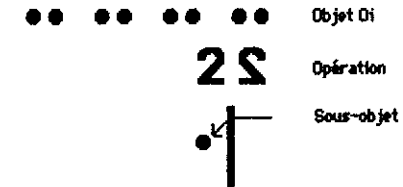
Avec le logiciel SIMUL, dans le contexte de commandes C (affichage par colonne) et M (affichage maîtrisé par le choix d'une option I), la sélection de la barre d'espacement comme choix I est une instruction "opération" qui s'applique sur le sous-objet constitué par la grille partielle présente à l'écran.

L'instruction "opération O_{pi} " signifie effectuer l'opération désignée ainsi par l'utilisateur sur les sous-objets en place, ceux-ci étant implicites.

un ou des sous-objets O_{bi} de O_i et des opérations O_{pi} relatives à O_i . **S6-[O_{bi} , O_{pi}]**

Ainsi, avec le logiciel SIMUL, l'affichage de la grille peut se faire grâce aux instructions "sous-objet ligne L" et "opération P" d'affichage continu.

Avec le logiciel FRISE, pour produire l'objet O_i ci-dessous, on peut donner les instructions "opération" et "sous-objet" indiquées.



L'explicitation du cas S6 est analogue à celle du système S4, avec cette différence: ici, les opérations O_{pi} , comme les sous-objets, sont désignées par l'utilisateur.

des sous-objets O_{bi} de l'objet O_i . **S7-[O_{bi}]**

Ainsi avec le logiciel LOGO, pour produire l'objet O_i ci-dessous, on pourra donner les trois instructions sous-objets: "segment (orienté)" [AV 15], "secteur angulaire (orienté)" [TD 90], "segment (orienté)" [AV 10].

Objet O_i

Avec le logiciel PICTOR, pour produire un objet O_i rectangulaire, on peut, ayant sélectionné le mode dessin, donner les instructions "sous-objet" que sont les désignations directes à l'écran des points du rectangle souhaité.

L'instruction "sous-objet" signifie strictement construire le sous-objet de O_i désigné ainsi par l'utilisateur. Elle n'inclut pas d'opérations implicites. L'utilisateur définit et produit l'objet O_i par emploi itéré d'un système S7 associé, le cas échéant, à un système de type S8. Avec ce type de système d'instructions, il doit expliciter à la machine les tâches inhérentes à la production de l'objet O_i .

□ des opérations Opi sur des sous-objets de Oi préalablement mis en place par l'utilisateur. Ss-[Opi]

Ainsi avec le logiciel PICTOR, pour produire un rectangle plein, on peut faire afficher tout d'abord à l'écran un rectangle vide (sous-objet), puis commander l'"opération" de remplissage en désignant son pictogramme sur la barre du menu et un point intérieur à la figure.

Les opérations numériques du logiciel LOGO constituent un autre exemple.

On notera qu'ici la désignation de l'opération ne comprend pas implicitement celle des objets sur lesquels elle s'applique – ce qui était le cas avec un système de type S5.

En conséquence, ici, la production d'un objet Oi relève de l'utilisation d'un des systèmes précédents (en particulier S7) conjointement à un système Ss. Toutefois, il ne s'agit pas du contexte S6: les opérations ne sont pas attachées à des objets dans un but défini par le logiciel, mais à des classes (les opérations numériques aux nombres, les remplissages aux domaines fermés, etc ...), sans objectif prédéfini par le logiciel.

TABLE DES MATIERES



La grille CONT	4
Commentaire des rubriques de la grille CONT	8
Descriptif de la typologie CONT	12
La grille COM	14
Commentaire des rubriques de la grille COM	18
Principes de schématisation de la structure des logiciels par rapport à la communication	25
Descriptif de la typologie COM	28
Arbre de premier classement selon les types de la typologie COM	34
Typologie générale CONT/COM	35
Vocabulaire d'usage et typologie	37
Huit systèmes d'instructions de production de logiciels à sémantique procédurale	38