

TYPOLOGIE DE LOGICIELS POUVANT IMPLIQUER
DES ACTIVITES MATHÉMATIQUES A L'ÉCOLE
ÉLÉMENTAIRE : QUELQUES RESULTATS

PAR F. TREHARD

cahier de
didactique des
mathématiques
numéro
16

O - INTRODUCTION

Dans ce cahier sont exposés des questions et des résultats établis dans le cadre d'une recherche* constituant une synthèse d'expériences d'activités informatiques en liaison avec la didactique des mathématiques à l'école élémentaire, expériences menées** à divers titres depuis 1981-82.

O.1. Situation de la recherche

Actuellement, le matériel d'analyse des logiciels (1° et 2° degrés d'enseignement) est limité.

On rencontre :

- des grilles d'examen des qualités techniques et/ou des conditions d'emploi des logiciels.
- des évocations (sans grilles d'analyse) de classements possibles des logiciels en référence à l'utilisation de spécificités de l'ordinateur, ou à des enjeux "pédagogiques" mal cernés, voire peu justifiables.
- des schémas descriptifs des logiciels d'enseignement programmé (schémas Crowdériens).

Ceci ne permet pas d'expliquer l'aval ou le qualificatif (didacticiel, imagiciel, logiciel-outil, micromonde, ...) accordés à certains logiciels.

Ceci n'autorise pas non plus un jugement rationnel des enjeux de l'usage d'un logiciel à des fins d'apprentissage mathématique à l'école élémentaire.

Dans la situation présente (implantations d'origines diverses de micro-ordinateurs dans les écoles), il y a un manque de fondements pour l'analyse comme pour la conception de produits informatiques.

Remarque :

La recherche entreprise concerne les logiciels de mathématiques pour le 1° degré. 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.

* Cette recherche fait l'objet d'une thèse nouveau régime en didactique des mathématiques.

** L'auteur est Professeur à l'Ecole Normale d'Institutrices de Paris, Animatrice à l'IREM Paris Sud).

0.2. Objectifs de la recherche

Objectif 1 : 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 (contenu, communication) et dégager une typologie à partir de ces grilles.

Cette typologie classera les logiciels indépendamment de leur utilisation effective pendant l'enseignement, ce qui explique son qualificatif "a priori"

Pour préciser, elle ne prendra 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.

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

Elle ne prétendra pas être à elle-seule un instrument de mesure de l'intérêt des logiciels.

Objectif 2 : Emettre des hypothèses relatives aux enjeux de l'utilisation effective (pendant l'enseignement) de logiciels de chaque type, aux rôles impartis à la machine, aux produits réformatiques souhaitables.

Il sera rendu compte de tests partiels.

Objectif 3 : Confronter les types dégagés à un vocabulaire informatique et didactique.

0.3. Remarque

Ce cahier rend compte essentiellement de l'objectif 1 de la recherche décrite ci-dessus.

1 - METHODOLOGIE RETENUE POUR ETABLIR LA TYPOLOGIE "A PRIORI".

1.1. Hypothèses

La méthodologie retenue repose sur 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 le 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

1.2. Définition avec le micro-ordinateur.

La méthodologie retenue se définit ainsi :

I (ou II)* 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 : typologie CONT

II (ou I)* 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 : Typologie COM

III Comparaison des typologies CONT et COM : recherche d'une typologie générale.

1.3. Remarques

- Choix des logiciels

Dans cette recherche, le terme "logiciel" désigne les logiciels d'exploitation et d'application, et ceci de façon délibérée.

* Les parties I et II sont indépendantes : l'ordre (I ou II) ne concerne que l'exposé.

Cette option peut se justifier à deux niveaux :

- . au niveau informatique, en référence à la signification du terme logiciel : "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 22.12.81, Ministres de l'Industrie et de l'Education Nationale).
- . au niveau utilisateur, en référence à sa position face à un système informatique : elle est indépendante a priori du logiciel. Pour l'utilisateur, il s'agit toujours moyennant un logiciel (d'exploitation ou d'application) d'utiliser un périphérique d'entrée et de lire - au sens large - un périphérique de sortie de la configuration dont il dispose.

Sur l'ensemble des neuf logiciels, on trouve :

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

Aucun logiciel n'est exclusivement ludique.

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

- Portée des résultats

Elle dépend de la représentativité des logiciels analysés.

D'autres logiciels pourraient confirmer les conclusions établies grâce aux neuf logiciels choisis.

2.1. Grille C O N T

	OUI	NON
<p>Niveau concerné</p> <p>CP</p> <p>CE</p> <p>CM</p>		
<p>Concepts mathématiques impliqués par le logiciel</p> <p>Type arithmétique</p> <p>Type géométrique</p> <p>Le logiciel permet-il d'aborder plusieurs concepts de même type ?</p>		
<p>Organisation du logiciel par rapport à un concept</p> <p>Le logiciel permet-il de traiter plusieurs problèmes relatifs à un même concept ?</p> <p>Un nombre fini déterminé ?</p> <p>Une infinité théorique ?</p> <p>Les problèmes sont-ils nécessairement isomorphes ?</p>		

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

La définition du contenu de l'activité

OUI | NON

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

L'enfant ou le maître peuvent-ils décider 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 logiciels ?

- 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 question conceptuelles ?

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

La validation de l'activité par rapport à son contenu

	OUI	NON
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		
. autre expression de la réponse de l'enfant		
- 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 ?		

2.2. 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 les dits logiciels.

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

Précisons certaines rubriques de la grille CONT :

- 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, fond du problème inchangé.

- 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.

- 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 conceptuelle 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, de maîtriser 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.
- . etc ...

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

- Rubrique : "Validation de l'activité par rapport à son contenu"

Les questions servant de titres aux 3 sous-rubriques correspondantes ont pour but de préciser dans quelle mesure l'enfant peut ne serait-ce que se poser le problème de la validation de son activité.

Pour la première question, il s'agit de savoir s'il y a un effet* qui permette à l'enfant de valider tout ou partie de l'activité.

* Il y a un "effet" s'il se passe quelque chose ayant un rapport avec le traitement qu'a effectué l'enfant du problème posé.

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.

Il peut y avoir effet au niveau de l'objet, mais éventuellement pas 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 pour 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.

Les deux sous-rubriques suivantes servent à examiner sur qui repose la validation du contenu de l'activité. Les questions ne se posent qu'en cas de réponse positive à la première question.

Elles ne sont pas exclusives; une réponse positive à l'une d'elles ne rend pas nécessairement l'autre sans objet.

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 :

Il peut penser valider, et en fait n'être pas en mesure de le faire réellement soit au niveau de l'objet^{*}, soit au niveau des

* Des enfants de CE 1 ont validé comme carré un graphisme produit sous LOGO par une série d'instructions où les rotations étaient de 89°.

contraintes de production de l'objet. Il peut aussi être tenter de valider sur autre chose que le problème posé (production d'un objet intermédiaire par exemple). Ceci explique la formulation "se sentir maître de la validation".

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, ...).

2.3. Descriptif de la typologie CONT

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

Celle-ci présente deux pôles (le type 1 et le type 2) et un 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 un 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 problème isomorphes sur un concept arithmétique ou géométrique. Le complément de définition porte 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 porte sur l'objet.

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

2.4. 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 1 font intervenir essentiellement des concepts arithmétiques et ceux du type 2 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 du type 1 ou 2.

2.5. Quelques questions

- Les faits suivants sont-ils contingents ?
 - . type 1 et concept arithmétique essentiellement.
 - . 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 ?

3.1. Grille C O M

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

Le terme "instruction" est précisé à l'explicitation de cette rubrique,
 en page 18.

Caractérisation :

Y-a-t-il :

- Ⓐ - instructions de définition de l'activité ?
- Ⓑ - instructions questions à l'utilisateur par rapport
 au problème qu'il traite ?
- Ⓒ - instructions de parcours du logiciel ?
- Ⓓ - 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 Ⓐ et Ⓑ -
 branchement dans le logiciel; voir explicitation
 page 25)

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

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

Nombre de type d'effets (1 ou 2)

Support réservé à l'évaluation

Graphique écran

sonore

mobile

Statut des entrées utilisateurs; degré de pro-
grammation laissé à l'utilisateur

(Pour des précisions sur les termes employés
ci-dessous, consulter l'explicitation de
cette rubrique page 25).

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

3.2. A propos des rubriques de la grille COM

Il est rappelé que la grille COM propose une analyse a priori des logiciels du point de vue de la communication proposé à l'utilisateur avec le micro-ordinateur, sans émission d'hypothèses relatives aux enjeux de situations qui emploieraient les dits logiciels.

Précisons certaines rubriques de la grille COM :

D'abord

" Les instructions pour la réalisation de l'activité"

Préalable :

On appellera "instruction" toute donnée brute du logiciel (y compris son environnement).

Cette définition implique en particulier que le terme "instruction" est envisagé d'un double point de vue :

- . instruction que l'utilisateur peut recevoir de la machine.
- . instruction que l'utilisateur peut donner à la machine.

- Rubrique "caractérisation"

Cette partie étudie l'objet de certaines instructions : il s'agit de celles qui permettent d'établir la communication, d'autoriser des formulations de la part de l'utilisateur*.

La rubrique "caractérisation" invite à classer ce genre d'instructions en quatre types (a - b - c - d).

* Il peut, conformément au sens donné au mot "instruction", exister d'autres genres d'instructions : en particulier, la machine peut émettre des instructions de gestion, de validation de l'activité. Il en sera question à la rubrique "évaluation des entrées-utilisateur" ? .

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	

Notes sur les types c) et d) :

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 multiplication 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 et contre-exemples :

Certaines instructions se présentent sous forme de phrases.

- 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.

Tous les codages ne sont pas des instructions au sens défini pour cette recherche.

Examinons les codages par lettres, nombres et mots "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 dans ces cas 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) .

Remarque : 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 étudiée par la rubrique "instructions".

- 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 eu regard aux rapports machine-utilisateur, elle n'entre pas dans l'activité proprement dite. L'entrée-utilisateur qui lui correspond ne sera pas non plus étudiée.

- Enfin l'utilisateur ne s'adresse pas à la machine nécessairement par instructions. Le cas des codages qui n'ont pas statut d'instructions a déjà été signalé. Il est possible de donner un autre exemple : - Calcule :

$$21 \times 12 = ?$$

"252" n'est pas une instruction, car ce nombre n'est pas une donnée brute du logiciel.

- Rubrique "support des instructions"

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

- Rubrique "validité des instructions"

Comme nous l'avons déjà évoqué, les instructions de type \textcircled{a} , \textcircled{b} et \textcircled{c} - repérage dans le logiciel sont généralement valides une fois, ou au mieux périodiquement sur le logiciel.

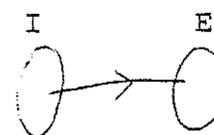
Les instructions de type \textcircled{a} et \textcircled{c} - branchement peuvent être données par l'utilisateur à la machine : elles peuvent être des entrées-utilisateur.

On remarque qu'elles peuvent avoir un plus grand domaine de validité.

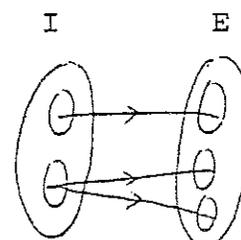
La rubrique "validité des instructions" étudie la validité des instructions qui peuvent être des entrées-utilisateur.

De façon plus précise, en appelant I l'ensemble des instructions qui peuvent être des entrées-utilisateur et E l'ensemble des entrées-utilisateur, on dira que :

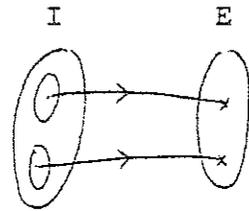
- \textcircled{a} - "toutes sont valides sur tout le logiciel" si on peut définir une relation dont le graphe soit $I \times E$



- \textcircled{b} - "certaines sont valides sur une partie du logiciel" si on peut définir une relation de graphe réunion d'ensemble $I' \times E'$, I' étant éventuellement égal à I et non réduit à un singleton, E' étant une partie stricte de E (la partie E' peut être une réunion d'ensembles disjoints).



- ⑦ - "certaines sont valides à un seul endroit du logiciel"
 si on peut définir une fonction de graphe réunion d'ensembles $I' \times \{e\}$, I' étant une partie stricte de I non réduite à un singleton.



Lorsqu'un même problème se répète plusieurs fois, il ne faut répondre aux questions de la rubrique "validité des instructions" que pour un exemplaire du problème.

Remarque :

L'étude de la validité des instructions renseigne sur la possibilité qu'a l'utilisateur d'effectuer une entrée avec des instructions diverses (I non réduit à un singleton).

Evoquons maintenant les rubriques concernant

" Les entrées-utilisateur "

Dans cette partie, le terme "écriture" est pris au sens large, c'est-à-dire qu'il peut s'agir d'une écriture réalisée au clavier ou de toute autre "écriture" produite avec un périphérique d'entrée de la configuration (périphérique vocal, analogique, stylo-optique, etc ...)

- Rubrique "caractérisation"

On regarde s'il y a coïncidence entre la notion d'instruction telle qu'elle a été définie précédemment et celle d'entrée-utilisateur.

En fait, une entrée sera le cas échéant :

- . une instruction de type @ ou © - 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 personnelles de l'utilisateur (voir pages 20 à 22).

Des entrées peuvent être des combinaisons d'instructions (examen d'une amorce de programmation).

Notons que certaines entrées qui suivent une instruction de type © peuvent être des instructions ou non.

- Rubrique "organisation"

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

Les deux premières questions font préciser s'il y a émission par la machine d'instructions suite à une entrée-utilisateur. Remarquons que les dites instructions peuvent faire partie de l'évaluation de l'entrée-utilisateur. Néanmoins, le problème général d'évaluation de l'entrée sera étudié par la rubrique suivante. Ici, il s'agit simplement de repérer la présence d'instructions données par la machine suite à une entrée de l'utilisateur.

La dernière question 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 de préciser si l'utilisateur est maître des entrées et de leur ordre (dans le cadre d'un respect des règles).

- Rubrique "évaluation"

On se place du point de vue de l'utilisateur.

Dans le cas où les entrées auraient été faites en mode différé, l'étude proposée par cette rubrique se situera à partir de l'exécution.

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.

Remarque :

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 @, et à un effet de type 2 si elle est de type ©.

- Rubrique "statut des entrées-utilisateurs; degré de programmation laissée 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 "caractérisation des entrées").
- . "commande de la machine de type pilotage", s'il est possible d'effectuer des entrées avec des instructions de type Ⓐ (voir les rubriques "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 Ⓐ sans provoquer d'évaluation (voir les mêmes rubriques que pour le point précédent).

L'analyse d'un logiciel selon cette 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 ©) 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 semble préférable de réserver le terme de "mode commande" pour ce qui concerne le contenu de l'activité. D'autre part, les entrées inhérentes au parcours d'un logiciel ont souvent autant le statut d'instruction que de réponse (au sens courant de ces termes). Leur formulation dans un logiciel permet de les apparenter à l'un ou l'autre mode (instruction de type © - branchement ou réponse), mais l'on pourrait changer cet état de fait.

Le mode d'expression du parcours d'un logiciel n'est donc pas fondamental. Il accentuera le cas échéant le caractère spécifique du logiciel. C'est en ce sens que l'on en tiendra compte.

3.3. 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 est accompagnée d'une schématisation relative à la structure des logiciels qu'il représente. Le principe de cette schématisation est donné en annexe (voir page 35).

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 :

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 la machine à l'écran.

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

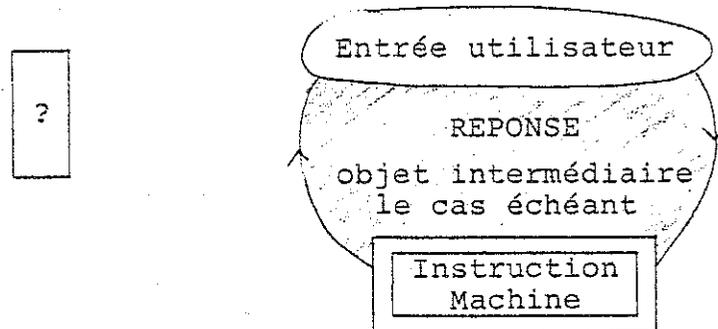
- En règle générale, les entrées utilisateurs ne sont pas des instructions. Elles ne peuvent l'être qu'exceptionnellement et concernent alors exclusivement le parcours du logiciel.

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 de 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.

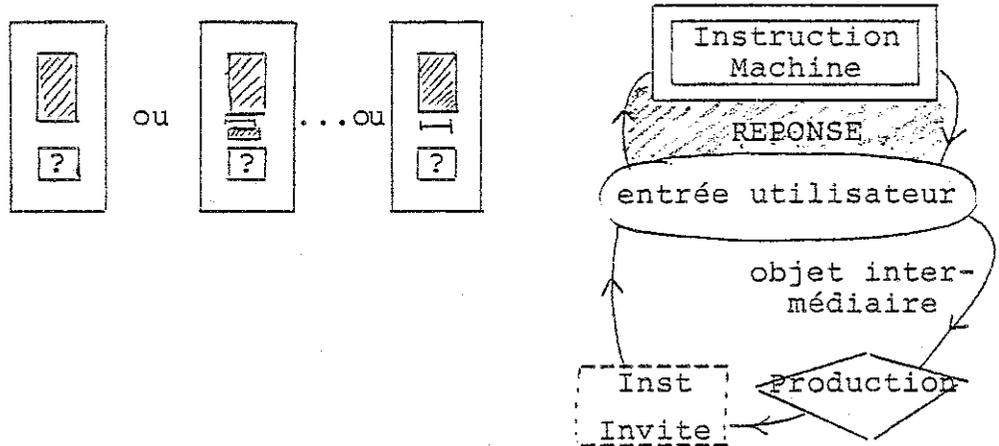
- Certaines entrées utilisateur sont des instructions. L'utilisateur est en mode commande de type pilotage pour la réalisation d'un objet intermédiaire et en mode réponse pour la réponse au problème traité. Suite à une entrée, il y a production et/ou instruction.
- Dans ce cadre, localement, l'utilisateur 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.

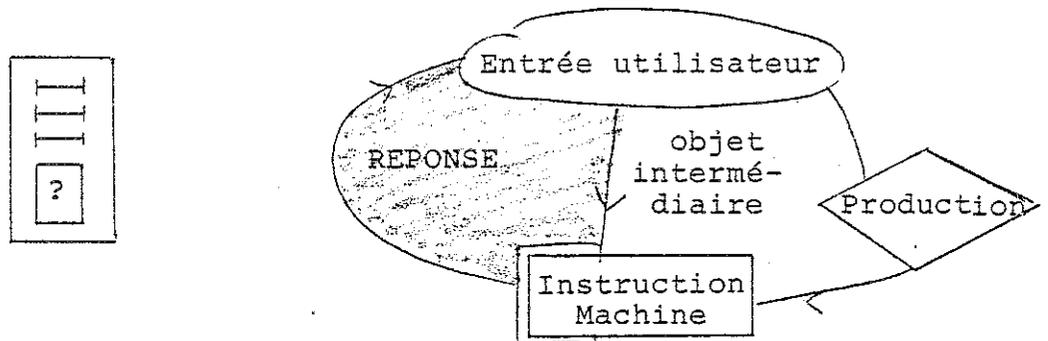
Notons qu'en sous-type I-II⁺ les deux genres de cycles sont nécessairement employés avec dominance des cycles production. En sous-type I-II⁻, les deux genres de cycle ne sont pas nécessairement employés. Il y a dominance des cycles réponse.

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 :

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.

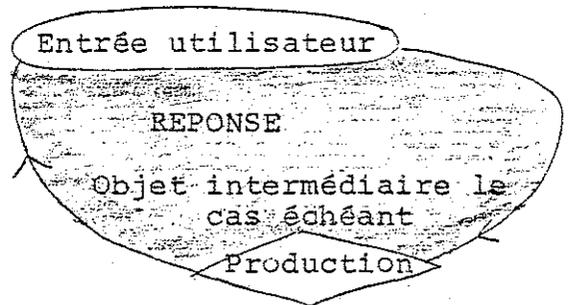
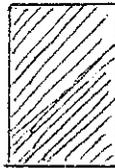
. Caractéristiques spécifiques aux sous-types II^- et II^+ :

- Les instructions sont données sur 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 du 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 :



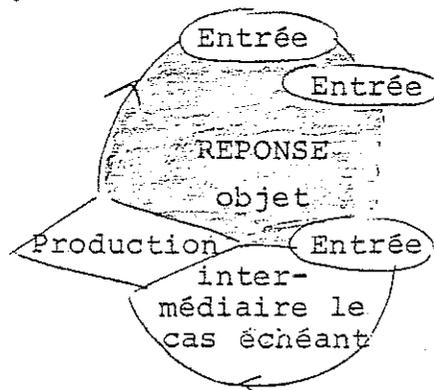
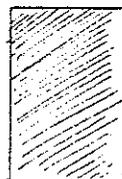
.. Caractéristiques spécifiques au sous-type II^+ :

Il existe un mode différé. L'anticipation est une donnée directement attachable au logiciel.

Schématisation :

Pour le mode pilotage, consulter celle du sous-type II^- .

Pour le 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.

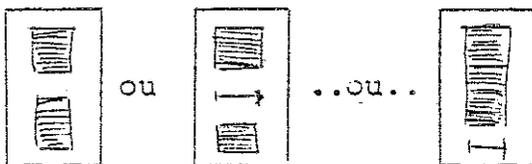
Suite à une entrée, il y a production et/ou instruction.

- 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

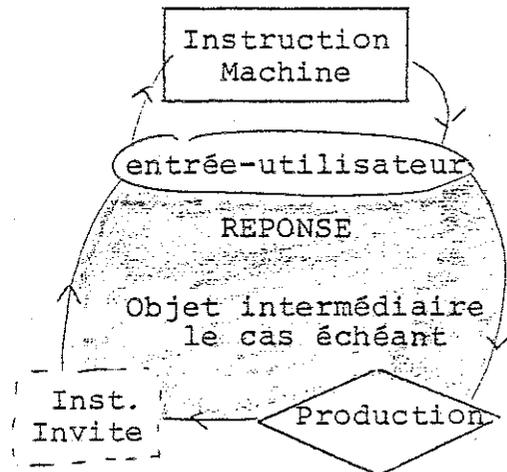
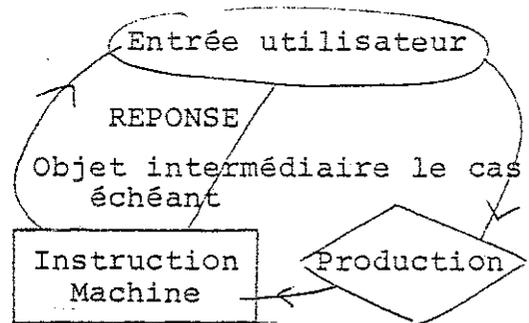
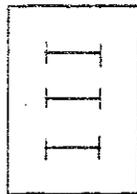


Schéma b :



3.4. Quelques remarques

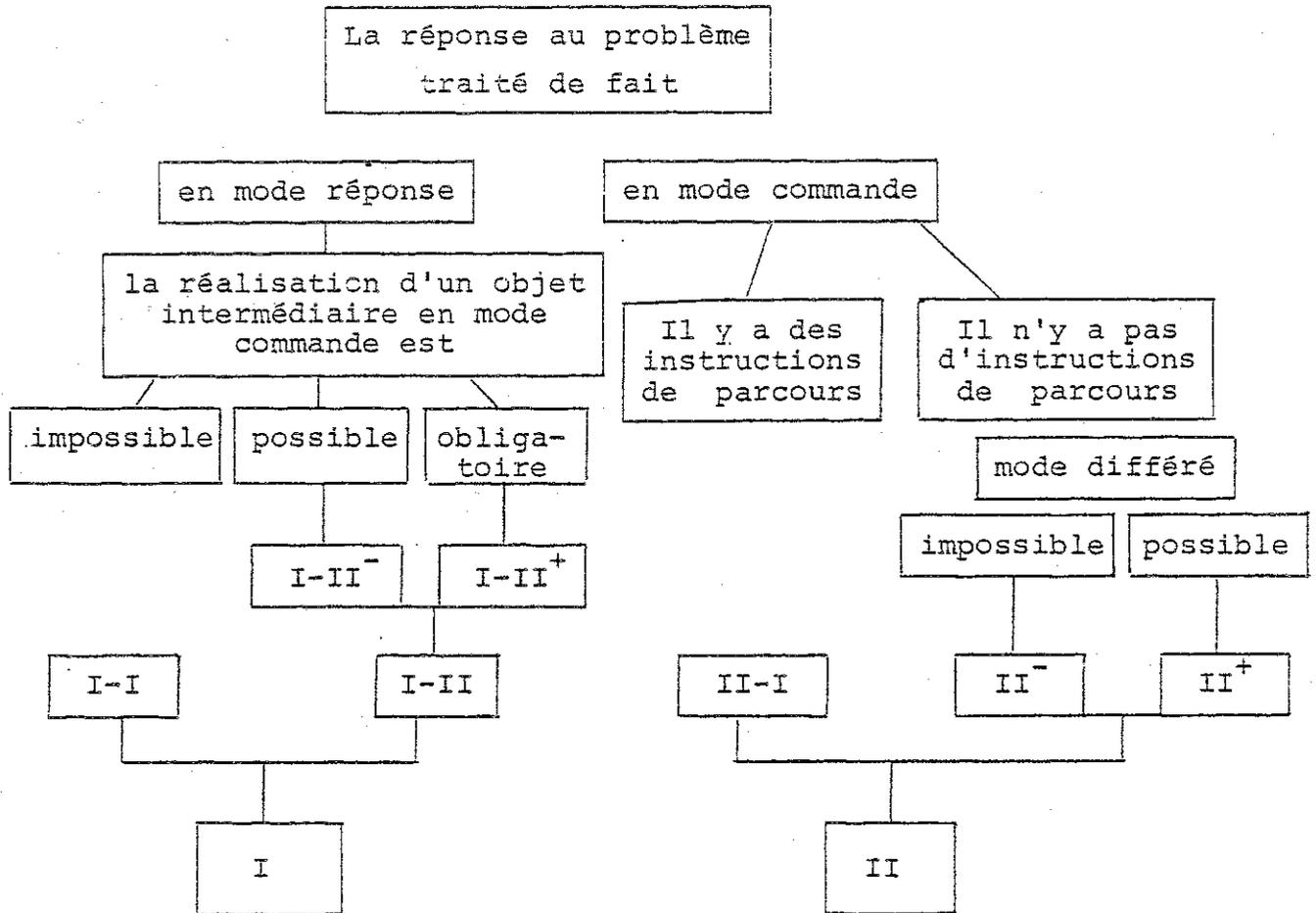
- Les logiciels de type I sont opposés aux logiciels de type II pour le mode d'émission de la réponse au problème traité. Les logiciels de type I-II permettent ou exigent la réalisation d'un objet intermédiaire en mode commande à l'inverse des logiciels de type I-I.

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 qui servira à effectuer un premier classement de logiciels. Pour assurer le classement, il conviendra néanmoins de faire une étude précise.

Arbre de premier classement :



- 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 II-I fonctionnent en type I (redondance) tandis que certains logiciels 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.

3.5. Quelques questions

- Comment l'utilisateur gère-t-il les divers genres d'entrée permises le cas échéant ?
- 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 ?

4. COMPARAISON DES TYPOLOGIES CONT ET COM : RECHERCHE D'UNE TYPOLOGIE GENERALE.

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 : *
- . type A
 - . sous type 1 I-I
 - . sous type 1 I-II⁻
 - . sous type 1 I-II⁺

* Pour retrouver les caractéristiques dans chaque cas, il convient de se reporter à l'exposé des typologies CONT (voir page 12) et COM (voir page 26) : les chiffres en écriture arabe renvoient aux types de CONT, ceux en écriture romaine aux types de COM

- . type B
 - . sous type 3.1 II-I
 - . sous type 3.2 II-I
- . type C
 - . sous type 2 II⁻
 - . sous type 2 II⁺

On peut se poser cette question : 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 ?).

L'examen des types A, B, C permet d'avancer que :

- 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 des sous-types différents pour le contenu (en B).

A contrario , au niveau des types mêmes, il semble qu'il y ait corrélation :

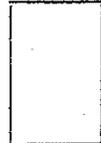
- le type 1 de CONT est attaché au type I de COM.
- les types 2 et 3 de CONT sont attachés au type II de COM; plus précisément, le type 2 de CONT est attaché aux sous-types II de COM sans caractéristiques I, tandis que le type 3 de CONT l'est au sous-type de COM avec caractéristiques I.

La question générale reste : quels sont les enjeux de situations utilisant des logiciels des divers types ?

Annexe : schématisation relative à la
structure des logiciels

- En faisant référence à la validité des instructions et à la caractérisation des entrées-utilisateur, on établit un schéma suivant ces principes :

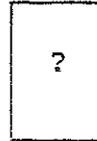
. Le logiciel est représenté par un rectangle :



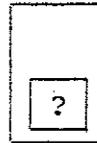
. On signale la présence :

- 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 :



- d'instructions valides sur tout le logiciel (voir page 22 type ④) couvrant toutes les entrées par le noircissement du rectangle représentatif du logiciel :



- d'instructions valides sur une partie du logiciel (voir page 22 type ⑤) par le noircissement de rectangles dans le rectangle représentatif du logiciel :



- d'instructions valides à un seul endroit du logiciel (voir page 23 type \odot) par le dessin de segments dans le rectangle représentatif du logiciel :



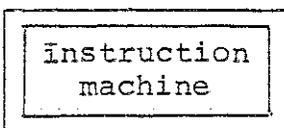
Les schémas comprennent généralement plusieurs des composantes qui précèdent.

- En faisant référence à l'organisation, à l'évaluation et au statut des entrées-utilisateur, on établit un schéma qui rend compte de ces divers points :

. L'entrée-utilisateur est notée par : entrée-utilisateur

. La présence d'une instruction machine est notée par : instruction
machine

S'il est possible que l'instruction machine soit une évaluation par rapport à l'attendu du conceptuel, on effectue un double cadrage :



. La présence d'une production suite et relative à l'entrée utilisateur est notée : Production . Ceci est caractéristique de logiciels définissant un mode commande.

. Les schémas s'organisent par cycles.

On signale le fait que la réponse s'effectue en mode commande ou non en incluant le mot REPONSE dans un cycle adéquat (cycle avec Production ou Instruction
machine)

Il en est de même pour la construction éventuelle d'un objet intermédiaire par l'utilisateur : on indique "Objet intermédiaire" dans un cycle approprié.