

# La transposition informatique. Note sur un nouveau problème pour la didactique

N. Balacheff

Abstract : we outline the main features of the *computational transposition*, a process likely to deeply transform the knowledge to be taught in the course of design and implementation of computer-based learning environments.

## 1. INTRODUCTION

La création des objets d'enseignement est le résultat d'un processus complexe d'adaptation des savoirs de référence aux contraintes d'enseignabilité et d'apprenabilité propres aux systèmes didactiques. Ce processus, la transposition didactique (Chevallard 1985), conduit à la création d'objets originaux par leurs caractéristiques et leurs fonctionnements propres. Le développement des technologies informatiques, leur introduction dans l'école et les lieux de formation, s'accompagne de phénomènes nouveaux du même ordre que ceux de la transposition didactique. Nous nous proposons dans cette note d'en dégager certaines des spécificités pertinentes du point de vue de la recherche en didactique des mathématiques. Aux contraintes de la transposition didactique s'ajoutent, ou plutôt se combinent, celles de modélisation et d'implémentation informatiques : contraintes de la *modélisation computable*, contraintes logicielles et matérielles des supports informatiques de réalisation.

Ce que l'on place habituellement sous le terme d'informatisation ne constitue pas une simple translittération, les environnements informatiques d'apprentissage résultent d'une construction qui est le lieu de transformations nouvelles des objets d'enseignement. Nous appelons *transposition informatique* le processus ainsi à l'œuvre.

## 2. LA TRANSPOSITION INFORMATIQUE

### 2.1. Multiplicité des lieux de représentation des connaissances

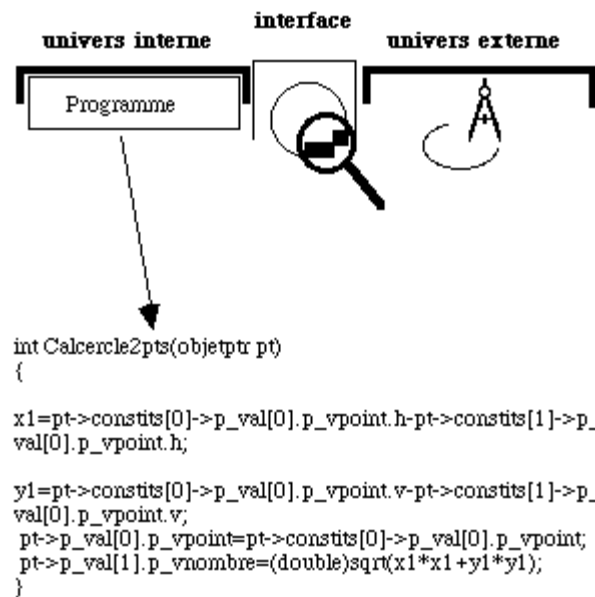
Les exigences propres à une modélisation computable tiennent à sa capacité de permettre la mise en œuvre d'un modèle symbolique de façon autonome par un dispositif informatique lieu de son implantation. Par dispositif informatique nous entendons ici le complexe formé par les matériels et les logiciels qui rendent opératoire l'*ordinateur*. Un

tel dispositif découpe le "monde" en trois régions qu'il sera commode pour la suite de bien distinguer (figure 1) :

- l'*univers interne*, fait de divers composants électroniques dont l'articulation et la mise en œuvre permettent le "fonctionnement" du dispositif informatique. Dans une simplification un peu rapide, mais qui ne réduira pas la portée de notre propos, nous considérerons qu'une représentation opérationnelle de cet univers est donnée par les langages de programmation.

- l'*interface*, lieu de la communication entre l'utilisateur humain et le dispositif informatique.

- l'*univers externe*, dans lequel se trouve l'opérateur humain et où lui sont éventuellement accessibles d'autres dispositifs (notamment en relation avec les connaissances en jeu dans le dispositif informatique). Nous n'aborderons pas ici les questions qui lui sont spécifiquement liées (voir à ce sujet, dans le présent volume, la communication de C. Laborde).



Evocation des représentations, à propos du cercle, dans les univers interne et externe, et à l'interface.

Fig. 1.

Prenons l'exemple du cercle évoqué par la figure 1. Dans l'univers interne, il est associé à des lignes de programme dans un langage donné, cette représentation subit des contraintes propres à ce langage et à la façon dont il est implanté. A l'interface, le cercle est un ensemble de pixels dans un état particulier, par exemple noirs sur un écran blanc. Suivant les caractéristiques de l'écran, le dessin du cercle produit par le dispositif

informatique sera plus ou moins perçu comme un cercle au sens commun. En fait, dans tous les cas, c'est un ensemble complexe de pixels qui ne renvoie à aucun assemblage de lignes au sens usuel de la géométrie. Dans l'univers externe, le cercle est une entité dont la nature est liée aux modes de représentation et de traitement disponibles, mais aussi aux classes de situations qui lui sont associées.

## 2.2. L'univers interne

Nous n'entrerons pas, ici, dans l'analyse des contraintes profondes de représentation et de mise en œuvre des langages de programmation utilisés (liées aux caractéristiques des systèmes d'exploitation). Par ailleurs, nous resterons dans le cadre des machines de la génération Von Neumann, les machines à venir renouveleront les questions mais ne supprimeront pas la pertinence de notre problématique. A titre d'illustration nous retiendrons, dans cette note, deux types de contraintes fondamentales au sens où les progrès techniques ne permettront pas de s'en affranchir totalement : granularité et compilation, représentation des données.

### 2.1.1. Granularité et compilation

La granularité est un concept de l'Intelligence Artificielle (I.A.) initialement très lié à la modélisation d'expertise en résolution de problèmes. La mise en œuvre de ces modèles exige leur description complète et donc le choix — dit de granularité — du niveau de termes primitifs. La compilation est un autre concept d'I.A., étroitement lié à celui de granularité. Il désigne un phénomène commun aux systèmes formels : un énoncé ne rend pas compte de l'histoire qui lui a donné naissance et en particulier des raisons de son existence. Dans le phénomène de compilation<sup>1</sup>, la formulation opératoire d'une règle occulte ses raisons théoriques — la règle est vraie du point de vue mathématique — et sa justification pragmatique — la règle a une utilité pour les problèmes considérés. Ainsi, par exemple, la règle de groupement additif exprimée dans le modèle d'APLUSIX (Nicaud 1993), logiciel tuteur en algèbre élémentaire :

Si E est de la forme  $B^P$  et B admet le facteur A avec le degré N et Q est un naturel canonique entre 1 et P alors E admet le facteur A avec le degré NQ

---

<sup>1</sup> Ce terme est repris du lexique informatique où il désigne la traduction d'un programme exprimé dans un langage de haut niveau en "langage machine", qui est accompagnée d'une perte des l'explicitation des structures. Ici, ce terme est utilisé métaphoriquement pour exprimer la perte particulière d'information liée à l'expression formelle d'une connaissance.

permet de déclencher un calcul mais ne rend compte ni de sa justification au plan mathématique ( $A^{NQ}$  divise  $A^{NP}$ ), ni des raisons de sa présence dans le modèle d'APLUSIX.

Les phénomènes de granularité et de compilation se conjuguent pour limiter de façon significative les possibilités d'analyse des processus issus de la computation. Ils sont une source de problèmes complexes pour le développement d'environnements qui auraient des capacités d'explication soit pour répondre à une demande directe de l'apprenant, soit pour soutenir une intervention sur l'apprentissage (Delozanne 1992).

### 2.1.2. La représentation des données

Le choix de représentations implique une structuration des données qui détermine des contraintes de traitement. Prenons un exemple en l'Algèbre : l'expression  $5x+3x(x-5)$  peut être considérée avec une structure de liste ou une structure d'arborescence liée à la hiérarchie des opérations. PIXIE (Sleemann 1982) et APLUSIX (Nicaud 1989) ont retenu respectivement la première et seconde structure. De ce fait, dans PIXIE, l'utilisateur peut exprimer la transformation (erronée du point de vue de l'algèbre usuelle) :  $5x+3x(x-5) \Rightarrow 8x(x-5)$ . En revanche, cette transformation n'est pas admissible dans APLUSIX, car  $5x+3x$  n'est pas une sous-arborescence de l'expression initiale. Cette différence doit être comprise à la lumière des objectifs différents de chacun de ces logiciels : APLUSIX vise les apprentissages de stratégie de résolution de problèmes et dispense l'utilisateur du niveau le plus élémentaire de calcul, PIXIE en revanche porte sur les manipulations élémentaires pour la résolution d'équations du premier degré.

## 2.3. L'interface

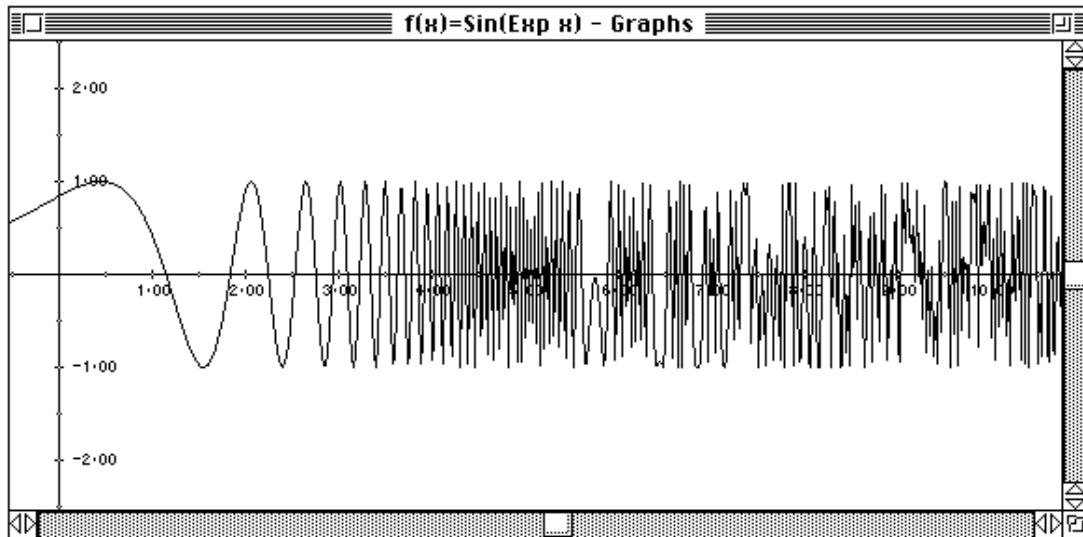
L'interface est le lieu de ce qui est souvent désigné comme la *réification* des connaissances : *visualisation* et *manipulation directe* d'entités abstraites donnant à voir des comportements évocateurs de leurs propriétés ; les phénomènes à l'interface, "offerts" à la perception donneraient ainsi une référence concrète aux concepts en jeu.

### 2.3.1. Visualisation

En mathématique, la visualisation est utilisée de façon naturelle pour les représentations graphiques de fonctions, en géométrie pour le tracé de figures, mais aussi pour exhiber des raisonnements sous la forme de graphes d'inférence.

Les systèmes de représentations ayant leurs propres caractéristiques, l'univers interne et l'interface combinent des effets générateurs et des phénomènes non intrinsèques aux entités représentées. L'exemple suivant illustre bien cette interaction. Les logiciels permettant le tracé de graphes de fonctions numériques s'appuient classiquement sur une

représentation approchée des nombres réels et sur des procédures de discrétisation permettant le *dessin* des graphes. Ce double "échantillonnage" produit inévitablement des effets de stroboscopie qui peuvent conduire à des effets pathologiques (figure 2).



Au-delà de  $x=3,5$ , la représentation du graphe de la fonction  $x \rightarrow \sin(\exp x)$  est "visiblement fausse". On ne peut utiliser les mêmes procédés graphiques pour tracer le graphe de  $x \rightarrow \sin x$  et celui de  $x \rightarrow \sin(\exp x)$ .<sup>2</sup>

Fig. 2.

Le micromonde de géométrie élémentaire Cabri-géomètre (Laborde J.-M. 1993) permet de visualiser les objets de la géométrie et de les manipuler. Ces objets ont une représentation interne (sur un modèle des nombres réels) et une représentation à l'interface. Cette dernière est constituée d'un pavage fini de pixels qui contraint la qualité perceptive des dessins. On peut espérer une meilleure résolution des écrans, mais quelle que soit la taille des pixels il restera à résoudre le problème de la désignation directe d'un objet : une décision doit être prise sur le "point montré" parmi un ensemble de points possibles. Cette décision peut au mieux prendre en compte une hypothèse sur l'intention de l'apprenant. Les limites de l'interaction fondée sur la perception sont susceptibles de conséquences sur les apprentissages, elles peuvent être aussi la source d'une problématisation féconde de concepts mathématiques.

### 2.3.2. La manipulation directe

<sup>2</sup> Cette illustration a été réalisée avec ANUgraph 2.0 (1991, Australian National University), un "grapheur" de très bonne qualité. En fait, des effets analogues seraient obtenus avec d'autres "grapheurs" ; le problème illustré ici a un caractère très général.

La manipulation directe, largement popularisée dans le domaine des jeux, ouvre l'accès à des interfaces dans lesquelles tout se passe *comme si* l'utilisateur pouvait avoir un accès direct aux objets, elle introduit une dimension perceptivo-gestuelle dans le répertoire des moyens possibles de communication. En mathématiques, son apport est très significatif pour la manipulation de représentations graphiques. Ainsi, dans Cabri-géomètre, la manipulation des figures permet d'observer leurs déformations et donc leurs éventuelles propriétés géométriques comme invariants de dessins dans la manipulation (Bellemain 1992).

Ces interfaces demandent la gestion des problèmes relatifs au comportement des objets au-delà des contraintes prises en charge par les modèles mathématiques sous-jacents. Par exemple, dans Cabri-géomètre, il faut décider du comportement d'un point quelconque d'un segment lors de la manipulation de l'une de ses extrémités. Ainsi, la question "quel est le lieu d'un point quelconque ?" prend curieusement du sens. Cette phénoménologie particulière de l'interface devient une caractéristique qui doit être prise en compte dans l'analyse du *milieu* dont le dispositif informatique permet la réalisation.

Les interfaces à manipulation directe conduisent à soulever un nouveau problème, celui de la relation entre le comportement des représentations et le comportement des modèles qui les sous-tendent. Un principe directeur de l'informatique graphique énoncé par Bresenham (1986), le WYSIWYE pour "what you see is what you expect", résume bien la position des concepteurs. Mais, le point de vue du didacticien est singulièrement différent. En effet, dans le contexte de l'apprentissage, l'interface n'est pas sous le contrôle théorique de l'utilisateur, elle peut au contraire devenir une référence relativement à laquelle la connaissance est construite.

### 3. CONCLUSION

Le problème de la représentation des connaissances est posé dans la littérature d'I.A. en termes de "fidélité" (Wenger 1987). Une telle problématique conduit à ignorer les limites autant que la réalité des bénéfices que l'on peut tirer des spécificités des représentations dans les dispositifs informatiques. Un glissement naturel dans ce contexte est de tenter de se soustraire à l'incontournable infidélité des représentations relativement à leurs "contenus" de référence (conséquence de leurs effets producteurs et réducteurs). Faire face à ce problème de transformation des connaissances dans le processus de représentation est d'autant plus essentiel que les phénomènes associés sont susceptibles de se combiner de façon complexe à ceux de la transposition didactique. A la problématique de la fidélité nous proposons de substituer celle du domaine de validité des représentations.

Le problème soulevé par la transposition informatique est celui du *domaine de validité épistémologique* des dispositifs informatiques pour l'apprentissage humain. La caractérisation d'un tel domaine est un problème complexe, nous y reviendrons ...

#### REFERENCES

- Bellemain, F. (1992) *Conception, réalisation et expérimentation d'un logiciel d'aide à l'enseignement de la géométrie : Cabri-géomètre*. Thèse de l'Université Joseph Fourier. Grenoble.
- Bresenham, J.E. (1988) Anomalies in incremental line rastering. *Theoretical foundation of computer graphics and CAD*. NATO ASI series vol. F40. Berlin: Springer Verlag.
- Chevallard Y. (1985) *La transposition didactique*. (nouvelle édition augmentée 1991) Grenoble : La Pensée Sauvage
- Delozanne, E.(1992) *Explications en EIAO: études à partir d'ELISE, un logiciel pour s'entraîner à une méthode de calcul de primitives*. Thèse de l'Université du Mans. Le Mans.
- Laborde J.-M. (1993) Intelligent microworlds and learning environments. in Laborde J.-M. (Ed.) *Intelligent Learning Environments: The case of Geometry*. Berlin : Springer Verlag. (sous presse).
- Nicaud, J.-F. (1993) *Modélisation en EIAO, les modèles d'APLUSOX*. Rapport de recherche n°859. Université de Paris Sud : laboratoire LRI.
- Sleeman, D.H. (1982) Inferring (mal) rules from pupils' protocols. *Proceedings of the European Conference on Artificial Intelligence*, Orsay, France. pp. 160-164.
- Wenger E. (1987) *Artificial Intelligence and Tutoring Systems*. Los Altos : Morgan Kaufmann Pub. Inc.