Chapter 6

AN INSTRUMENTAL APPROACH TO MATHEMATICS LEARNING IN SYMBOLIC CALCULATOR ENVIRONMENTS

Luc Trouche
LIRDEF, LIRMM & IREM
Université Montpellier II, France
trouche@math.univ-montp2.fr

Abstract: A rapid technological evolution (Chapter 1), linked to profound changes within the professional field of mathematics (Chapter 3), brings into question the place of techniques in mathematics teaching (Chapter 5). These changes have created serious difficulties for teachers; obliged to question their professional practices, they make different choices regarding integration of new technologies and techniques (Chapter 4), choices that are linked to their mathematical conceptions and to their teaching styles.

In this chapter, we place ourselves on the side of the students. We have already seen (Chapter 1) that they seem to adopt the new computing tools faster than the institution. In this chapter, we study more precisely their learning processes related to their use of symbolic calculators.

First of all, we pinpoint the didactic phenomena taking place in the experiments; subsequently, we suggest a new theoretical approach aimed at giving a better description, for each student, of the transformation of a technical tool into an instrument for mathematical work.

Key words: Computational transposition, Instrumentation and instrumentalization process, Instrumented technique, Operational invariants, Schemes.

1. DIDACTIC PHENOMENA APPEARING IN FIRST EXPERIMENTS

Research studies on symbolic calculators as environments have been conducted in France since 1995. These were preceded by research on calculator integration (from 1980), and by research on DERIVE software integration (institutionally supported in France from 1991). These studies have revealed many didactic phenomena. Artigue (1997) distinguishes two interrelated classes of phenomena: those linked to knowledge transposition and those linked to students' adaptation to new environments.

1.1 Didactic phenomena linked to processes of knowledge transposition

These processes are linked to *computational transposition* (Box 6-1), described by Balacheff (1994) as "work on knowledge which offers a symbolic representation and the implementation of this representation on a computer-based device".

Artigue (1997) brings out two phenomena linked to these processes:

- the phenomenon of *pseudo-transparency*, linked to the gap between what a student writes on the keyboard and what appears on the screen (a gap arising from differences between two representation modes, internal and interface):

> [To enter (a+2)/5], some students, having correctly written a couple of parentheses around (a+2), are surprised to see a screen display without parentheses. They wonder if their production is correct, or not. Parentheses appearing and disappearing seems to be a mysterious game they can't understand, precisely because they have not mastered parentheses techniques.

- The phenomenon of *double reference*, linked to the double interpretation of tasks, depending on the work environment (paper-and-pencil or computerized). Artigue (ibid.) evokes in the following terms the rational factorization of $x^n - 1$ in a 11^{th} grade class, with DERIVE software (Box 5-2):

> In a paper-and-pencil environment, polynomial factorizations are linked, at this school level, to the search for real roots (…). In the software environment, these rational factorizations come first from factorizations in Z/pZ (…): the factorization by (x – 1) for example is obtained only if n is a prime number and the factorization by (x – 1) (x + 1) only for n = 4 (…). Students choosing the machine interpretation have much greater difficulties in producing conjectures.

Box 6-1.

Computational transposition
(Balacheff 1994)

Balacheff defined *computational transposition* in the following terms:
"A representation of the world is not the world itself. This now largely shared assertion can be taken as a commonplace. Nevertheless, … to understand it, we have to go further and consider that a representation is not an approximation (i.e. a simplification) of its object in order to re-present it. Each representation has properties which come both from modeling choices and from chosen semiotic modes. These properties have, a priori, no connection with the represented world. Moreover, as a material device, a computer imposes a set of constraints which themselves will impose an appropriate transformation allowing the implementation of the adapted representation.
I will name as *computational transposition* this work on knowledge which offers a symbolic representation and the implementation of this representation on a computer-based device, in order to *show* knowledge or to manipulate it. In a learning context, this transposition is particularly important. It implies indeed a contextualization of knowledge, with possible important consequences for learning processes".
Balacheff distinguishes constraints linked to the internal universe of a machine (for example, the program for representing a circle, Figure 6-1) from interface constraints (for example, the screen representation of a circle distorted by pixellation).

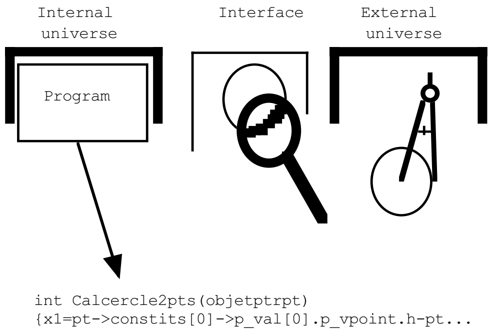

Figure 6-1. Circle computational transposition

These phenomena point to the importance, for learning, of precisely delimiting the domain of *epistemological validity* of an artifact, i.e. to characterize the objects it gives access to and to identify its semiotic and functional characteristics.

1.2 Didactic phenomena linked to students' processes of adaptation to environments

Experiments also reveal processes of adaptation to environments: in these processes, the constraints and potentialities of computerized environments play a determining role.

1.2.1 Perceptual adaptation processes

First of all these processes are linked to the potentialities of calculators as regards *visualization*, in terms of graphical as well as algebraic *representatives* (Schwarz & Dreyfus 1995). The influence of 'direct'[1] perception is most widely noted in the graphic frame. We showed (Guin & Trouche 1999), for example, that answers to the question "Does the function f, defined by $f(x) = \ln x + 10 \sin x$, have an infinite limit as x tends to $+\infty$?" depend strongly on the working environment (even though elementary theorems make it possible to answer *yes* to this question).

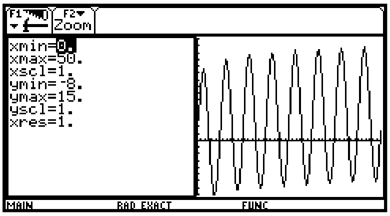

Figure 6-2. A graphic representation of the function f: $x \mapsto \ln x + 10\sin x$

In a graphic calculator environment, 25% of students answered *no*, appealing to the oscillation of the observed graphic representation (Figure 6-2); in a paper-and-pencil environment, only 5% of students answered *no*.

The importance of framing perception in algebraic terms was pointed out by Artigue (1997); students, in a DERIVE environment, had to explain how to move from the equation $2x - 5y = 8$ to the equation $12x - 30y = 48$, then $6x - 15y = 24$.

> On first passage, the teacher gave an indication: multiply by 6. This will favor a solution based upon formal analogies which are essentially perceptual (…). [Students] suppose that 6 has been obtained by dividing, but wonder how this division could be done. After some hesitation, they decide to try using DERIVE, to do something which according to them, "*would probably give nothing, but trying costs nothing*": they enter the two equations under division and ask DERIVE to simplify. DERIVE answers with: 3 = 3, which amuses students ("*it's trivial!*") but also intrigues them. They do not try to understand this answer, but decide to do the same thing with the first equation. This time, $(12x - 30y = 48) / (2x - 5y = 8)$ gives 6 = 6.
> As DERIVE answers 6 = 6, when the teacher's answer is 6, so, in the second case, when DERIVE answers 3 = 3, the right answer must be 3.

Thus, perceptual adaptations can come into play within both graphical and algebraic frames, but this does not guarantee the establishment of relationships between these two frames: Dagher (1996) shows that frequent use of software allowing algebraic and graphical representations of functions to be manipulated does not necessarily help students to build an efficient articulation between these two frames.

Perceptual adaptations are also linked to potentialities for *animation*. We have pinpointed numerous manifestations of this in calculator environments; for example (Trouche 1995), students had to find a parabola tangential to three given lines (Figure 6-3). To perform this task, students tested diverse parabola equations. In order to check if their parabola was a correct solution, they zoomed in on the contact point between a line and the parabola. They supposed this contact to be 'good' if, after several zooms, curve and line appeared confounded on the calculator screen. At the end of this work, the teacher asked: "*How can a tangent to a curve be defined?*"

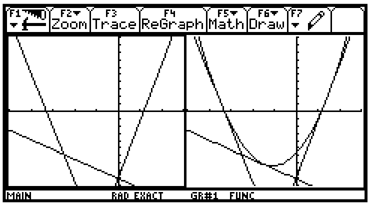

Figure 6-3. Search for a parabola tangential to three given lines

The first answer proposed from the class was: *"a line is the more tangential to a curve the more common points it has with it"*. This definition does not correspond with any taught knowledge acknowledged by the institution; it is the simple translation of students' observation of a 'good' contact between a curve and a line on a calculator screen composed of pixels (Figure 6-3). This is a visualization effect, linked to the computational transposition (particularly, here, the constraints of discrete traces, Box 6-2).

In tackling the same task, the strategy of finding by trial and error a curve of equation $y = ax^2+bx+c$ also leads to the construction of knowledge (related to the roles of the coefficients a, b and c). In this context, students' activity is essentially based on observing the displacement of curves through modification of coefficients in their equation. Thus, for c, students claim: "*when c increases the parabola goes up, when c decreases the parabola goes down"*. Even after a clarification from the teacher ("*c is the y co-ordinate of the intersection of the curve with the y axis"*), it is often the first interpretation that is memorized: when the teacher asks about the sign of the coefficient c in the equation of the specified parabola (Figure 6-4), some students answer that c is equal to zero in the case of the left parabola ("*the parabola is at the level zero"*) and c is negative in the case of the right parabola ("*the parabola is underneath the x-axis"*), although c is strictly positive in both cases.

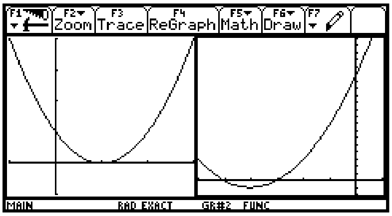

Figure 6-4. Two unknown parabola equations

This reaction shows the importance of movement for students' perception and description of what appears on a screen (what students remember is more the *change* between two images, rather than the properties of each image. The same reasoning in terms of animation (more precisely, the possibility of moving a point on a curve thanks to the *Trace* command) could explain students' conception of a function graph, in a calculator environment, as the trajectory of a moving point, rather than as a set of points whose coordinates are (x; f(x)).

Box 6-2.

Constraints of discrete plots and some consequences
(Guin & Trouche 1999)

Many phenomena arising in relation to the graphical interface of calculators are linked to the presentation of discrete plots on a screen composed of a finite number of pixels.

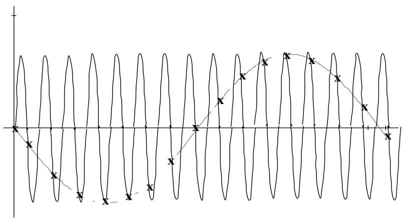

Figure 6-5. A diagram showing the consequence of a discrete plot: an usual period for sine function

For example, if the represented function is periodic (sine function, Figure 6-5) and if the distance between two computed points is close to its period, the two computed images will also be close together. As the calculator joins these two points, the oscillation between these points will not be shown.

1.2.2 Phenomena linked to the organization of students' work

The *multiplicity* of *easily available* commands has effects on the economy of students' work.

Students carry out trials and tests without paying attention to their organization and control. They hope that, within a reasonable time, they will obtain something interesting.

> Observations show that these fishing behaviors can be productive for students, often more productive than more reflective behaviors available to them. The low cost of these trials and their productivity tend to discourage retroactive approaches, involving looking back and modifying accordingly, generally considered as essential in generating the cognitive adaptations hoped for (Artigue 1997).

In the same category, Defouad (2000) pinpoints a *zapping* phenomenon (which consists of quickly changing graph window, without having time to analyze each of the representations obtained), an *oscillation* phenomenon (students oscillating between several techniques and strategies) and an *over-checking* phenomenon (students carrying out multiple checks, using all the means provided by the calculator).

We have also noted similar phenomena, in a calculator environment (Trouche 1997):

- a phenomenon of *automatic transportation*: students enter all the problem data into the calculator, and then look for the command which could give the solution directly:

> [A student] studies a positive sequence u(n) converging toward 0. He wants to determine the value of n from which u(n) will be smaller than 10^{-10}. He takes his calculator, enters u(n) in the sequence editor, enters 10^{-10} in the window setting, as "nmax". Then he wonders what is the right key he has to press in order to have the result?

- a phenomenon of *localized determination*, linked to the difficulty of moving from one *register* to another one (Duval 2000; Guin & Trouche 2002, p.158) and of changing application on a symbolic calculator. It consists of repeating the same type of technique, within the same calculator application, making some adjustments, even if this type of technique does not appear relevant.

> To answer the question "Are there some power functions with curves tangential to the curve of the exponential function?", some students tried in succession (x^2, then x^3, then $x^{2.1}$, then $x^{2.2}$…). Each curve was tested through successive zooms. During the whole activity (taking one hour), the same type of approach was repeated.

While looking back is exploited effectively (unlike in fishing behavior), work remains confined to a single graphical application, with a double consequence:

- firstly, for problem solving: doing mathematics often requires changing one's point of view. Keeping the same point of view, using a single technique, often does not allow a given problem to be solved;
- secondly, for building knowledge: working in a single register, representing a mathematical object in a single form, does not make it possible to form a complete notion of this object.

These phenomena generally appeared in long-term experiments, where students had calculators at their disposal (both at school and at home). These parameters are probably important, facilitating the appropriation of the calculator by students and the stabilization of techniques to perform given tasks. This necessity of taking into account the potentialities and constraints of new tools led us to study appropriation and utilization processes. More generally our interest in *mediation* linked to the learning process (Vygotsky 1962) led us to seek new theoretical approaches, which would yield better understanding of the role of material and symbolic *instruments* within mathematical activity.

2. A NEW APPROACH IN ORDER TO UNDERSTAND AND DESCRIBE NEW PHENOMENA

Recent work in the field of cognitive ergonomics has provided theoretical tools allowing a better understanding of processes of appropriation of complex calculators. Verillon and Rabardel, dealing with training in general (1995) propose a new approach, which essentially distinguishes an *artifact* from an *instrument*:

- an artifact is a material or abstract object, aiming to sustain human activity in performing a type of task (a calculator is an artifact, an algorithm for solving quadratic equations is an artifact); it is *given* to a subject;
- an instrument is what the subject *builds* from the artifact.

This building (Figure 6-6), the so called *instrumental genesis*, is a complex process, linked to characteristics of the artifact (its potentialities and constraints) and to the subject's activity, her/his knowledge and former work methods.

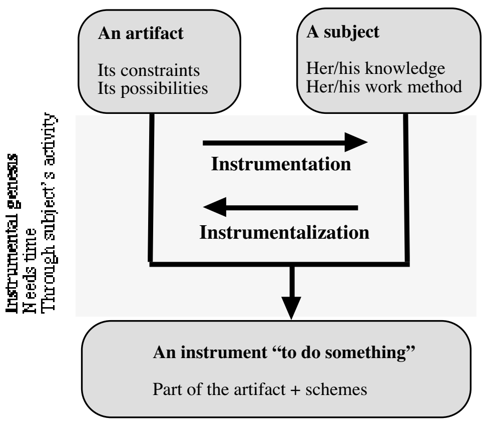

Figure 6-6. From artifact to instrument

This schema rests on some fundamental ideas:

- an artifact *partially* prescribes the user's activity, through its constraints and potentialities;
- instrumental genesis is a process (therefore *needs time*) and has two components, the first one (*instrumentalization*) directed toward the artifact, the second one (*instrumentation*) directed toward the subject;
- a subject builds an instrument *in order to perform a type of task*; this instrument is thus composed of both artifact (actually *a part* of the artifact used to perform these tasks) and subject's *schemes* (Box 6-4) allowing her/him to perform tasks and control her/his activity.

We are going now to make these ideas more precise in the context of symbolic calculators.

2.1 Analyzing constraints and potentialities of symbolic calculators

Computational transposition and design choices produce constraints in a symbolic calculator which Balacheff classifies as internal constraints and interface constraints (Box 6-1).

Regarding general relationships with artifacts, Rabardel (1995) distinguishes three types of constraint: *existence mode constraints*, linked to properties of the artifact as a cognitive or material object, *finalization constraints*, linked to objects it can act on and to transformations it can carry out, and, lastly, *action prestructuration constraints*, linked to prestructuration of the user's action.

Concerning symbolic calculators, we have used (Trouche 1997) both Balacheff's and Rabardel's typologies, distinguishing *internal constraints* (identified as existence mode constraints), *command constraints* (linked to the existence and the nature of specific commands) and *organization constraints* (linked to ergonomic questions, particularly keyboard and menu organization).

Defouad (2000) notes some shortcomings in this typology:

- internal constraints do not cover all existence mode constraints (for example, the nature of the calculator screen is not an internal constraint, but an existence mode constraint);
- all the constraints actually prestructure the user's activity (and not only organization constraints);
- this typology does not take into account various *information levels*: information introduced by the user at the interface, information accessible at the interface, but not open to transformation by the user, and information not accessible at the interface;

- it does not take account of *syntax constraints*, even though these can be decisive when introducing information at the interface.

Box 6-3.

Internal constraints of one graphic calculator

(Bernard & al 1998)

The authors studied internal constraints of the TI-82 calculator. Figures 6-7 and 6-8 show one illustration, linked to implemented algorithms for approximate computation: while the limit at 0 of the given function is 1/6, the table of values and graph of the function give first a value close to 1/6, then, as x approaches 0, produce some oscillations, and finally seem to give, as the function limit, the value 0.

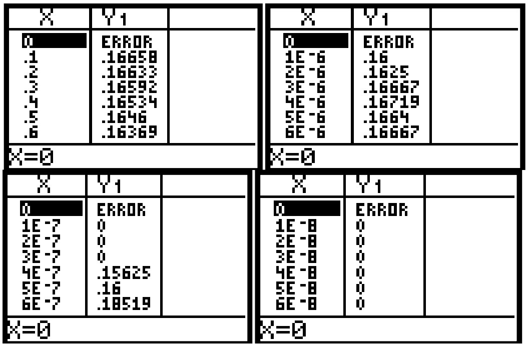

Figure 6-7. Numerical observation of the function f: $x \mapsto \dfrac{x - \sin x}{x^3}$ near 0

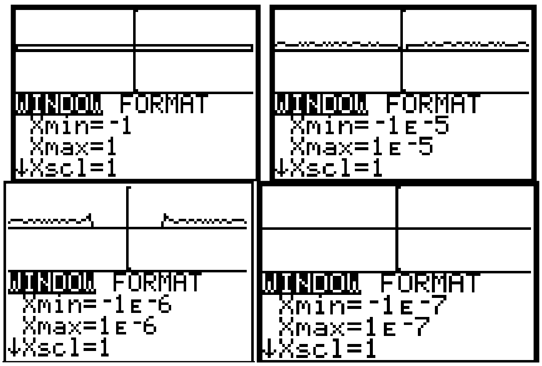

Figure 6-8. Graphical observation of the function f: $x \mapsto \dfrac{x - \sin x}{x^3}$ near 0

Taking account of these remarks, we make precise three types of constraint, all serving to prestructure the user's action and related to a type of task:

- *internal constraints* (in the sense of physical and electronic constraints) intrinsically linked to material. They are linked to information which the user cannot modify, whether accessible or not. They strictly shape action. They include for example processor characteristics, memory capacity (Box 6-3) and screen structure, composed of a finite number of pixels (Box 6-2);

- *command constraints* linked to the various commands in existence and their form (including syntax). They are linked to information accessible at the interface which the user can sometimes modify:

> Example 1: the *Range* application of the calculator allows the viewing window to be fitted to a graphic representation of the function. The choices open to the user are relatively free: s/he can choose Xmin and Xmax (but not Xmax smaller than Xmin). The graphic representation of the function which is obtained through setting these values provides feedback allowing a better fitting window to be found.
>
> Example 2: some calculators (Texas Instruments symbolic calculators) require the use of parentheses when computing function values (sin(2), log(3), etc.). This is not the case for other calculators (Casio symbolic calculators) which accept entries such as sin2, or log3: these different design choices can have consequences for students' conceptions about functions.

- last, *organization constraints* linked to keyboard and screen organization, i.e. to available information and command structure.

> Example 1: designer choices related to functions (the naming of commands, means of accessing them and their placing within a menu) give a particular point of view on available objects (Appendix 6-1). These choices are linked to an ergonomic study of users' needs, and, at the same time, they favor a particular form of tool use.
>
> Example 2: the placing of the symbol "∞" is not neutral. On Texas Instruments calculators, this symbol is directly given by a keystroke (and it can be manipulated as a number or a letter). On Casio symbolic calculators, it is available only in the CAS application. These different approaches can instill different relationships with this symbol ∞, and, beyond, with the notion itself (Appendix 6-1).

It is possible to discuss the placing of a given constraint into one of the three defined types. But this interest in typology is not strictly in partitioning constraints; it is rather in making easier, for teacher as well as for researcher, an *a priori* analysis of different ways proposed for performing tasks with an artifact. Distinguishing these three levels allows this analysis to be organized in *a given mathematical context* (Box 8-5, for such an analysis of limit computation). Particularly, distinguishing an elementary level of *command* constraints and a more complex level of *organization* constraints permits a distinction, within students' activity, between a level of *gesture* and a level of *technique*.

Analyzing calculator constraints shows clearly that it presents mathematical knowledge in a particular way: "These tools wrap up some of the mathematical ontology of the environment and form part of the web of ideas and actions embedded in it" (Noss & Hoyles 1996). A user is thus not 'free' to use, as s/he wants, a given tool: "This use is, relatively, prestructured by the tool" (Luengo & Balacheff 1998). These constraints do not necessarily lead to impoverishment of activity: by taking in charge part of the work, by favoring exploration in various registers (Yerushalmy 1997), tools open new ways for conceptualization. It is indeed difficult to separate potentialities on the one hand and constraints on the other: both are intimately linked, each facility offered presses the user to realize one type of gesture rather than another.

2.2 Understanding two components of instrumental genesis: one directed toward the artifact, the other directed toward the subject

Instrumental genesis (Figure 6-6) is a process of building an instrument from an artifact. It has two closely interconnected components:

- the *instrumentalization* process, directed toward the artifact;
- the *instrumentation* process, directed toward the subject.

The instrumentalization process, directed by the subject, involves several stages: a stage of *discovery* and *selection* of the relevant keys, a stage of *personalization* (one fits the tool to one's hand) and a stage of *transformation* of the tool, sometimes in directions unplanned by the designer: modification of the tool bar, creation of keyboard shortcuts, storage of game programs, automatic execution of some tasks (the web sites of calculator manufacturers or the personal web sites of particularly active users often offer programs for functions, methods and ways of solving particular classes of equations etc.). Instrumentalization is a process of differentiation as regards the artifacts themselves:

- differentiation regarding the calculator's contents: in making comparisons between students' calculators it is possible to identify differences (from both quantitative and qualitative points of view) between the various programs stored;
- differentiation regarding that part of the artifact mobilized by the subject (for some students, a very small part of calculator, for others a large one).

Instrumentalization is the expression of a subject's specific activity: what a user thinks the tool was designed for and how it should be used: *the elaboration of an instrument takes place in its use*.

Instrumentalization can thus lead to enrichment of an artifact, or to its impoverishment.

Instrumentation is a process through which the constraints and potentialities of an artifact *shape* the subject. As Noss & Hoyles (1996) note: "Far from investing the world with his vision, the computer user is mastered by his tools". This process goes on through the emergence and evolution of schemes (Box 6-4) while performing tasks of a given type. We will study an example of such processes in the following section. As instrumentalization processes, instrumentation can go through different stages. Defouad (2000) analyzes these processes of evolution for students who, after using graphic calculators, then move on to use symbolic calculators (TI-92). He distinguishes two main phases, first an *explosion* phase and second, a *purification* phase:

> At the beginning of instrumental genesis, the student's work seems to be at a crossroads, as if s/he was looking for an equilibrium between her/his former techniques and strategies (linked to graphic calculators) and various new possibilities opened up by TI-92 calculators and the evolution of classroom knowledge. We call this phase an *explosion phase*, as new strategies and techniques appear to burst out; it seems to be characterized by *oscillation*, *zapping* or *over verification* phenomena (§ 1).
> Progressively, students enter into a second phase we call a *purification phase*, where machine use tends to an equilibrium, in the sense of the stabilization of instrumented strategies and techniques. This phase often goes with a fixation on a few commands (and such choices could be different, according to each student).

Box 6-4.

Schemes and conceptualization
(Vergnaud 1996)

Vergnaud distinguishes:
- Conceptions: "one can express them by sequences of statements whose elements are objects, monadic or polyadic predicates, transformations, conditions, circumstances, forms"…
- Competencies: "one can express them by actions judged adequate for the treatment of situations".

He introduces the *scheme* concept, allowing relationships to be established between conceptions and competencies. A scheme is an invariant organization of activity for a given class of situations. It has an intention and a goal and constitutes a functional dynamic entity. In order to understand its function and dynamic, one has to take into account its components as a whole: goal and subgoals, anticipations, rules of action, of gathering information and exercising control, *operational invariants* and possibilities of inference within the situation.
Vergnaud names as *operational invariants* the implicit knowledge contained within schemes: *concepts-in-action* are concepts implicitly believed to be relevant, and *theorems-in-action* are propositions believed to be true. He distinguishes theorems-in-action and concepts-in-action ("truth is not the same thing as relevance"), but insists on their deep links ("theorems-in-action cannot exist without concepts-in-action, as theorems cannot exist without concepts, and vice-versa"). These operational invariants occupy a central place in this frame as: "two schemes are different as soon as they contain different operational invariants".

To better understand the complexity of these two processes, let us make two elements precise:

i) Instrumental genesis is a process of building an instrument, from an artifact, by a subject. This instrument is built from a part of the initial artifact (modified through the instrumentalization process) and through schemes built in order to perform *a type of task* (Box 5-1). A complex artifact such as a symbolic calculator will thus give birth, for a given student, to *a set of instruments* (for example an instrument for solving equations, an instrument for studying function variation, etc.). The articulation of this set is a complex task (Chapter 8, § 2).

ii) Instrumental geneses have both *individual* and *social* aspects. The balance between these two aspects depends on:

- material factors (it is quite obvious that the 'intimacy' of calculator screens favors individual work whereas computer screens allow common work by small student groups);
- the availability of artifacts (sometimes, they are available only at school, sometimes they are lent for the whole school year, sometimes they are students' property);
- the way in which the teacher takes these artifacts into account (Chapter 8, § 2).

Moreover artifacts are mediators of human activity and activity mediated by instruments is always *situated* (Chapter 8, § 2).

Chacon & Soto-Johnson (1998) analyze some effects of these variables on students' behaviors and on their relationships with artifacts: when calculators or computers are available only from time to time, students often develop a critical attitude toward technology; indeed they are sometimes quite confused (because learning in the two environments -- computerized and 'classical' -- is not the same) and frustrated (computers are not available outside laboratory scheduled work).

2.3 Understanding different levels and different functions of instrumented action schemes

Rabardel (1995) introduced the notion of the *utilization scheme* of an artifact to describe a scheme operative within activity mediated by an artifact and distinguishes two such sorts of schemes:

- *usage schemes*, "oriented toward the secondary tasks corresponding to actions and specific activities directly linked to the artifact";
- *instrumented action schemes*, whose "significance is given by the global act aiming to carry out transformations on the object of activity".

All are partially *social schemes*, as their emergence comes, in part, from a collective process involving artifact users and designers. Schemes of usage and instrumented action are deeply linked. A scheme of instrumented action aims to perform a given task. It includes *operational invariants* (Box 6-4).

One can consider instrumented action schemes as a set of usage schemes. Understanding the function of a usage scheme requires it to be considered not in isolation, but as a component of an instrumented action scheme involved in performing a given task.

2.3.1 Usage schemes and gestures

We define a *gesture* as the observable part of a usage scheme. For example, we illustrate (Trouche 2000) the importance of a particular gesture, *approximate detour;* it consists of a combination of keystrokes which results, when working on a symbolic calculator in exact mode, in an approximate value of a symbolic expression. It is not a simple gesture, only oriented toward calculator management: beyond (or psychologically underneath) this gesture, there is a usage scheme, with associated knowledge. Looking for this knowledge involves considering the gesture not as an isolated act but as integrated within an instrumented action scheme employed by the student in order to resolve given tasks.

We identified (Trouche 1996) the three main schemes of instrumented action in which approximate detour appears as those of solving equations, computing integrals, and computing limits.

The observation of students' work shows rules of action, of gathering information, of exercising control (Box 6-4):

- for some students, the approximate detour has always a *determination function* (the approximate value obtained is considered as *the* value);
- for other students it has always an *anticipation or checking* function (obtaining an approximate value may be a step in the process of seeking an exact value).

In other words, approximate detour contributes to building different kinds of knowledge about, say, the real numbers.

2.3.2 Instrumented action schemes and instrumented techniques

One can describe human activity (and students' activity in particular) in terms of *techniques* (Box 5-1), i.e. sets of gestures realized by a subject in order to perform a given task. When a technique integrates one or several artifacts, we will speak of an *instrumented technique*. Instrumented technique is thus the observable part of an instrumented action scheme. For example, an instrumented technique which can be *described* in this way (Trouche 2001) is one for limit computation, in a symbolic calculator environment, as presented by a teacher (Figure 6-9).

Its presentation is made as a tree. In general, such a tree can be:

- more or less 'vast' (in the sense of the number of calculator applications used, of the number of frames evoked, etc.). We can observe in this case that the numerical frame is not used (for numerical observations, for example);
- more or less dense (in the sense of the *granularity* of prescribed gestures). In this case, use of the calculator in order to split the problem is not indicated.

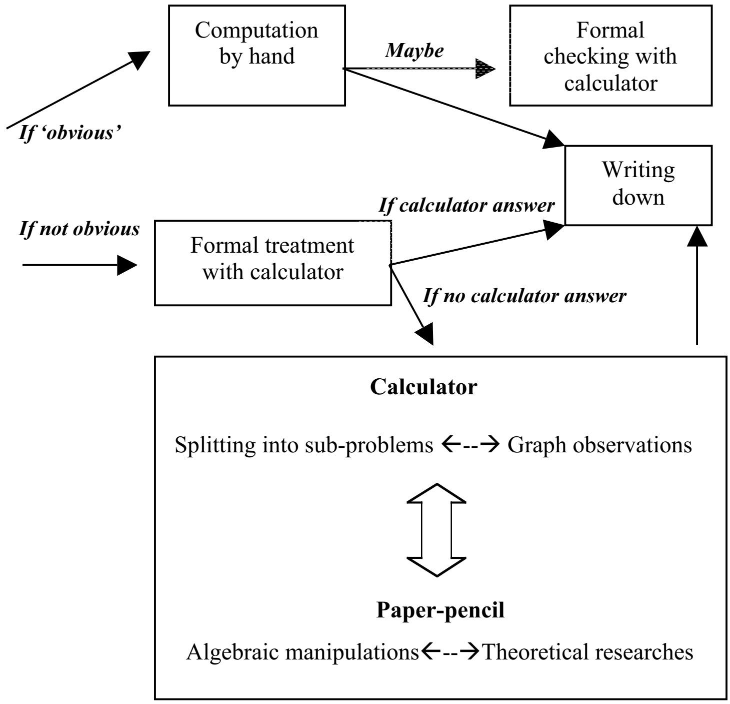

Figure 6-9. An instrumented technique for limit computation, as seen by a teacher

An instrumented technique can be *taught*, but what is taught is not necessarily what students *learn*: the gap between instrumented techniques as taught and as practiced may be important (Appendix 6-2, which shows two students' very different work within the same class and for the same taught instrumented technique).

Describing activity in terms of instrumented action schemes calls for consideration of operational invariants (Box 6-4). A scheme is an observer's construction from the different activity traces of a subject (gestures, anticipations, inferences, etc.). Let us illustrate this, for the same student, in two different environments. He is a student from an experimental class (Trouche 2000), working first for three months with graphic calculators, then for six months with symbolic calculators. The task consists in studying the following question: *"has the given function an infinite limit as x tends to +∞?"*.

Graphic calculator environment

If one only described the gestures of the instrumented technique, one would say that the student takes his calculator, 'enters' the function to be studied in the function editor, uses numerical applications to observe the function behavior for large values of the variable, and infers the answers by observing the values taken by f(x) as x takes large values.

If one wants to look at the instrumented action scheme (Figure 6-10), one will search for the operational invariants *guiding* this technique. Searching for them depends on observing the student performing other tasks of the same type and asking him to justify his answers. This student explains he does not use the graphical application, because defining a 'right' window for graphing the function on a large scale is too difficult. Therefore he uses the calculator table of values, and, as far as he can, infers the function behavior. He thus concludes, in the following cases: *"if f(x) is much greater than x, or if the function increases with great speed it is okay. On the other hand, if the function starts to decrease or oscillate then it is no good"*. Consequently one can hypothesize that the student's scheme integrates theorems-in-action of the following type *"if f(x) takes much larger values than x, then the limit of f is infinite"*, *"if the function increases very strongly, then the limit of f is infinite"*, *"if the limit of f is infinite, then f is necessarily increasing"*. From all these properties emerges a concept-in-action of the type: *"f has an infinite limit means that, when x is large, f(x) is very large, increasingly large"*[2].

Symbolic calculator environment

If one describes only instrumented technique, one will say that the student takes his calculator and applies the limit command to the given function.

Concerning the instrumented action scheme (Figure 6-10), there is, compared to the graphic calculator environment, an apparent *simplification*: less effort while manipulating the artifact (the only effort is a syntactic effort of writing a correct command) and less effort of explanation (since the software ensures the correctness of results, any justification of a result, even when required by the teacher, appears less necessary). The instrumentation process leads here to a simplification of the scheme, accompanied by an impoverishment of the operational invariants. To the question: *"what is the meaning of the function having an infinite limit?"* the student, in a graphic calculator environment, gave an answer related to the concept-in-action which we evoked above; four months later, in a symbolic calculator environment, he could not give any definition any more: the function limit did not have any other existence than as a product of the software symbolic application, as a response to a computation command. There was a *vanishing* of the concept. Vanishing does not mean disappearing: the limit conception moves from a *process result*, in a graphic calculator environment, to an *operation result*, in a symbolic calculator environment.

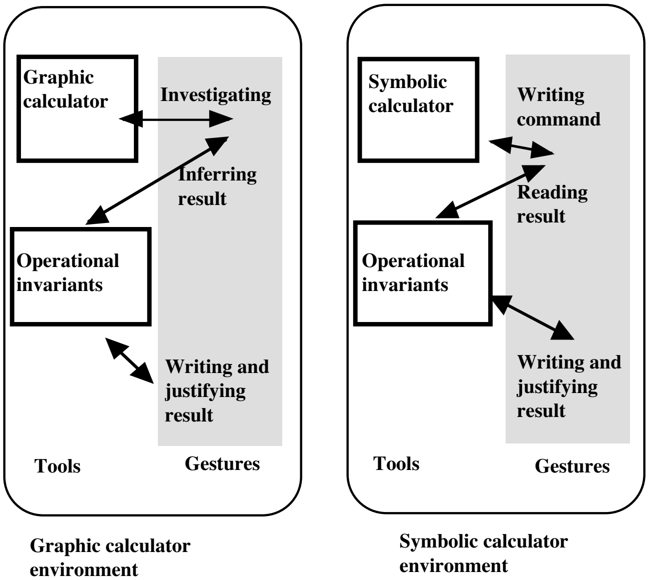

Figure 6-10. Evolution of limit computation action schemes, from a graphic calculator environment to a symbolic one

As we can see for these two instrumented action schemes, there is a dialectic relationship between operational invariants and realized gestures:

- operational invariants *guide* gestures: in the first case, they guide gestures through the investigation process, the inference process and the justifying process. In the second case, they guide gestures of writing a command, reading a result, and (although weak) a process of justifying. In the graphic calculator environment, the mobilization of operational invariants requires an important cognitive effort (one has to evaluate if the x values are large 'enough' and if f(x) is large 'enough'). In the symbolic environment, the cognitive effort is not of the same nature: it is not related to a search process, but only to a control of syntax (here we speak of a particular student; amongst other students, we observed other schemes, Appendix 6-2);
- at the same time, activity, through gestures, *institutes* operational invariants: "From successive approximations, the hand finds the right gesture. The mind registers the results and infers an efficient gesture scheme. Gesture is a synthesis." (Billeter 2002). Operational invariants appear as an abstraction of what is judged an apt gesture. Then, because

operational invariants enable a task to be performed, their field of operationality and validity will naturally spread.

A schema expresses this dialectic between action and conceptualization (Figure 6-11).

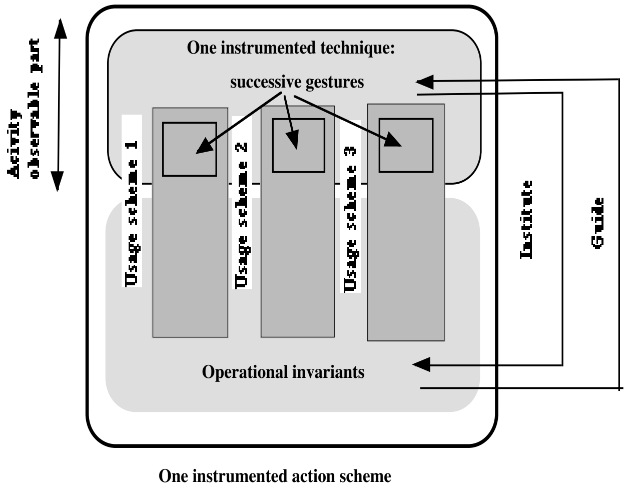

Figure 6-11. Relationships between scheme and technique, gesture and operational invariants

This study of schemes of instrumented action leads to two conclusions:

- the first one relates to the two instruments successively built by the same student: it clearly appears that *extension (or complexification) of an artifact can go with a reduction (or an impoverishment) of the corresponding instrument built by a subject*[3]*;*
- the second one relates to the method of studying instrumented action. The study of instrumented action schemes requires studying, beyond the techniques themselves, their epistemic, heuristic and pragmatic functions (Box 5-1). It requires analysis of the student's activity in more depth: over time, in order to pinpoint regularities, and with regard to the student's discourse, in order to pinpoint the justification offered for gestures. These regularities of activity and justifications of gestures allow hypotheses about operational invariants to be formulated.

Having a good knowledge of calculator constraints and more precise ideas on students' operational invariants may give teachers some means to orient their mathematics lessons:

- choosing situations which help students to master concepts (something which cannot be realized for an isolated concept, but only in the frame of a conceptual field, Box 6-5). This question will be studied in Chapter 9;
- taking into account the artifacts available in the learning environment in order to favor social aspects of schemes. We will see this point in Chapter 8.

Box 6-5.

Operational invariants, concepts and conceptual fields
(Vergnaud 1996)

A *concept* acquires its sense through several situations and phenomena and, thus, has its roots in several categories of operational invariants. Besides, it becomes fully a concept only through articulation of its properties and of its nature, in a mathematical wording where it has the status either of predicate or object.

This idea leads to the definition of a *concept* as a triplet of three sets:

- a set of situations which give sense to the concept;
- a set of operational invariants through which such situations are treated;
- a set of language and symbolic representations which allow the concept to be represented.

A concept cannot be built in isolation. It has to be studied as an element of a larger set which Vergnaud names a *conceptual field* (for example, the limit concept has to be built within the conceptual field of calculus). One can define a conceptual field in a twofold manner:

- as a set of situations needing to be progressively mastered and a closely interconnected range of concepts, procedures and symbolic representations;
- as the set of concepts which ensure a mastering of these situations.

APPENDICES

Appendix 6-1
Some organization constraints for two symbolic calculators

What are, for two different symbolic calculators, the organization constraints related to study of functions?

Computation

TI-92 (Texas Instruments)	Algebra FX 2.0 (Casio)
The same application (available on the keyboard: *HOME*) allows numerical and formal, approximate and exact computation. Within this application, the user can choose the computation mode s/he wants. This mode is indicated at the foot of the screen.	Two different computation applications: first *(NUM)* for approximate computation, second, *CAS*, for formal and exact computation. The computation mode does not remain indicated on the screen when calculating something.

Two different design choices:
- for Casio, approximate mode appears privileged through the first application proposed, *NUM*, only allowing approximate computation;
- for TI, the two modes are placed on the same plane. A keystroke combination (*approximate detour*, § 2.3.1) allows the shift from an exact to an approximate result (from example from 1/3 to 0.33333). Both exact and approximate values can coexist on the same screen.

Between the two cases, a different relationship to numerical approximation is favored.

Combination of commands

TI-92 (Texas Instruments)	Algebra FX 2.0 (Casio)
One can combine approximate and symbolic computation, writing for example the following command: *Approximate value(limit f(x), x, a)*.	It is impossible to combine approximate and symbolic computation commands.

These different design choices could have consequences, for example on the conceptualization of the limit notion.

Graphical and numerical analysis of functions

TI-92 (Texas Instruments)	Algebra FX 2.0 (Casio)
GRAPH application (allowing graphically representing function) and *TABLE* application (allowing obtaining table of values) are accessible on the same level on the keyboard.	Opening *GRAPH* application gives access to several menus. One of these menus contains the *TABLE* application. This application thus appears to be included within the *GRAPH* application.

Two different design choices:
- for Casio, graphical representation (compared to table of values) appears privileged;
- for TI, these two types of representation are on the same level;

In these two cases, it is not the same graphical/numerical articulation which is favored.

Appendix 6-2
Two instrumented techniques for computation of limits, in a symbolic calculator environment, for two 12th grade students

(Trouche 2001, p.16)

One can see below the gap between the instrumented technique which is taught (Figure 6-9) and the two students' techniques.

Students had to determine the limit as x tends to $+\infty$ of the function f defined on $]0; +\infty[$ by $f(x) = \dfrac{\sqrt{x} + \cos x}{x + \sin x}$.

NB. The TI-92 symbolic calculator does not "know" this limit (Box 8-4, § 2.1, p.222), but students could use some basic theorems to derive this limit as equal to 0.

<u>Student 1</u>

He first defines the f function for the calculator (Figure 6-12, next page) "*then I could avoid having to write this complex thing several times*". Calculator answer: undef.

> "*Oh, these functions sine and cosine often cause trouble when looking for limits, I need to get rid of them*".

On his paper, he bounds sine and cosine as lying between – 1 and +1, and then bounds the f function, for $x > 0$:

$$\frac{\sqrt{x}-1}{x+1} \le f(x) \le \frac{\sqrt{x}+1}{x-1}$$

He uses his calculator to find the limits of the left and right function: 0.

> "*According to the theorem about limits and inequalities, I can say that my function f has also 0 as a limit*".
>
> "*Let us have a look at the graphs of the three functions.* He graphs the three functions: *"the function f is well bounded by the two others in the neighborhood of* $+\infty$*"*.

Then: "*I can also change the variable*". On paper:

$$X = \sqrt{x}, \quad f(X) = \frac{X+1}{X^2+1}$$

> "*I can use the theorem about the polynomial functions, or do some factorizations and use the theorems about limits and operations*":

On paper again:

$$\frac{X+1}{X^2+1} = \frac{1+\dfrac{1}{X}}{X+\dfrac{1}{X}}$$

End of the work (one hour): paper-and-pencil and calculator approaches articulated, a work in multiple-registers (algebraic and graphical studies), expression and construction of knowledge about limits, a rich limit scheme.

The tool complexity is mastered and contributes to enrich the instrumentation process and to build an efficient instrument for study of function limits.

<u>Student 2</u>

He uses the limit command of the CAS application, applied to the given function.

Calculator answer: undef.

> *"Oh, I made a mistake in writing the command!*"

He writes again, same calculator answer.

"Oh, I am useless, I'll have to try again" (writing the function $\frac{\sqrt{x}+\cos x}{x+\sin x}$ takes a lot of time). Same calculator answer…

"Oh, I have understood, the calculator doesn't know the f function, I have to define it!".

He defines the function f (Figure 6-12).

Again the command limit, again the answer undef.

New perplexity, and new idea: "*when a limit isn't defined, it is sometimes possible to look at the left, or at the right of the point. So I am going to look at the right of* $+\infty$*, so I will go as far as possible*" (Figure 6-12). Still answer undef.

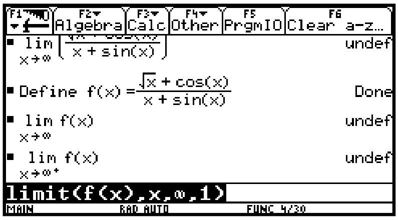

Figure 6-12. Calculator screen copy of student 2

At last, he breaks down the problem into sub-problems, looking for the limits of $\sqrt{x}$ (*"It works, I obtain* $+\infty$ *as a limit!"*) and of sinx and cosx (*"that is the problem: these two functions have no limit, that is the reason why my function f has not limit"*).

End of the work (1hour 30minutes): no paper used, work in a single register (no numerical nor graphical studies), no idea of the function behavior, a quite weak scheme for studying limits.

The complexity of the tool does not contribute to assist the student's activity or build an efficient mathematical instrument.

NOTES

1. "We don't see only forms, but also meanings" (Wittgenstein, in Bouveresse (1995)).
2. This concept-in-action appears close to a *kinematic point of view* on function limit (Box 8-4).
3. This situation is not necessarily linked to a given environment. In the same class, we showed (Trouche 1996) the existence of very different processes for other students: different forms of instrumentalization developed (storage into calculator of the main theorems related to function limits, of specific programs for computation of limits, etc.) and instrumentation becoming richer with the shift from graphic to symbolic calculator environment, through use of a great diversity of applications (Appendix 6-2).

REFERENCES

Artigue M. (1997) Le logiciel DERIVE comme révélateur de phénomènes didactiques liés à l'utilisation d'environnements informatiques pour l'apprentissage, *Educational Studies in Mathematics* 33(2), 133-169.

Balacheff N. (1994) Didactique et intelligence artificielle*, Recherches en Didactique des Mathématiques* 14(1/2), 9-42.

Bernard R., Faure C., Noguès M., Nouazé Y. & Trouche L. (1998) *Pour une prise en compte des calculatrices symboliques en lycée*. Montpellier: IREM, Université Montpellier II.

Billeter J.F. (2002) *Leçons sur Tchouang-Tseu*, Editions Allia.

Bouveresse L. (1995) *Langage, perception et réalité*, Editions Jacqueline Chambon.

Chacon P.R. & Soto-Johnson H. (1998) The Effect of CAI in College Algebra incorporating both Drill and Exploration, *The International Journal of Computer Algebra in Mathematics Education*, 201-216.

Duval R. (2000) Basic Issues for Research in Mathematics Education, in T. Nakahara & M. Koyama (Eds.), *Proceedings of PME 24* (Vol. 1, pp. 55-69). Hiroshima: Hiroshima University.

Dagher A. (1996) Apprentissage dans un environnement informatique: possibilité, nature, transfert des acquis, *Educational Studies in Mathematics* 30, 367-398

Defouad B. (2000) *Etude de genèses instrumentales liées à l'utilisation de calculatrices symboliques en classe de première S,* Doctoral Thesis. Paris: Université Paris VII.

Guin D. & Trouche L. (1999) The complex process of converting tools into mathematical instruments: the case of calculators, *International Journal of Computers for Mathematical Learning* 3*,* 195-227.

Luengo V. & Balacheff N. (1998) Contraintes informatiques et environnements d'apprentissage de la démonstration en géométrie, *Sciences et Techniques Educatives* 5(1), 15-45.

Noss R. & Hoyles C. (1996) *Windows on Mathematical Meanings, Learning Cultures and Computers* (pp. 153-166). Dordrecht: Kluwer Academic Publishers.

Rabardel P. (1995) *Les hommes et les technologies, approche cognitive des instruments contemporains*. Paris: Armand Colin.

Schwarz B.B. & Dreyfus T. (1995) New actions upon old objects: a new ontological perspective on functions, *Educational Studies in Mathematics* 29(3), 259-291.

Trouche L. (1995) E pur si muove, *Repères-Irem* 20, 16-28.

Trouche L. (1996) *Enseigner les mathématiques en terminale scientifique avec des calculatrices graphiques et formelles*. Montpellier: IREM, Université Montpellier II.

Trouche L. (1997) *A propos de l'enseignement des limites de fonctions dans un "environnement calculatrice", étude des rapports entre processus de conceptualisation et processus d'instrumentation*, Doctoral Thesis. Montpellier: Université Montpellier II.

Trouche L. (2000) La parabole du gaucher et de la casserole à bec verseur: étude des processus d'apprentissage dans un environnement de calculatrices symboliques*, Educational Studies in Mathematics* 41(2), 239-264.

Trouche L. (2001) Description, prescription, à propos de limites de functions, in J.B. Lagrange & D. Lenne (Eds.), *Calcul formel et apprentissage des mathématiques* (pp. 9-26). Paris: INRP.

Vergnaud G. (1996) Au fond de l'apprentissage, la conceptualization, in R. Noirfalise & M.J. Perrin (Eds.), *Actes de l'Ecole d'Eté de Didactique des Mathématiques* (pp. 16-28). Clermont-Ferrand: IREM, Université Clermont-Ferrand II.

Verillon P. & Rabardel P. (1995) Cognition and artifacts: A contribution to the study of thought in relation to instrument activity, *European Journal of Psychology of Education* 9(3), 77-101.

Vygotsky L.S. (1962) *Thought and Language*. Cambridge, MA: MIT Press.

Yerushalmy M. (1997) Reaching the unreachable: technology and the semantics of asymptotes, *International Journal of Computers for Mathematical Learning* 2, 1-25.