Computer-based learning environment in mathematics

Nicolas Balacheff, James J. Kaput

COMPUTER-BASED LEARNING ENVIRONMENTS IN MATHEMATICS

Nicolas Balacheff & James J. Kaput

Authors version of: Nicolas Balacheff & James J. Kaput (1997) Computer-Based Learning Environments in Mathematics. In: Bishop A. (ed.) *International Handbook in Mathematics Education* (pp. 469-501). Dordrecht: Kluwer Academic publisher.

ABSTRACT: This chapter attempts to set a perspective on where interactive technologies have taken us and where they seem to be headed. After briefly reviewing their impact in different mathematical domains, including arithmetic, algebra, geometry, statistics, and calculus, we examine what we believe to be the sources of technology's power, which we feel is primarily epistemological. While technology's impact on daily practice has yet to match expectations from two or three decades ago, it's epistemological impact is deeper than expected. This impact is based in a reification of mathematical objects and relations that students can use to act more directly on these objects and relations than ever before. This new mathematical realism, when coupled with the fact that the computer becomes a new partner in the didactical contract, forces us to extend the didactical transposition of mathematics to a computational transposition. This new realism also drives ever deeper changes in the curriculum, and it challenges widely held assumptions about what mathematics is learnable by which students, and when they may learn it. We also examine the limits of Artificial Intelligence and microworlds and how these may be changing. We close by considering the newer possibilities offered by the internet and its dramatic impact on connections among learners, teachers, and the immense resources that are becoming available to both. Our conclusion is that we are very early in the technological transformation and that we desperately need research in all aspects of teaching and learning with technology.

KEYWORDS: mathematics education, computers, microworld, artificial intelligence, mathematical modeling, distance learning, computational transposition, algebra, geometry, statistics, arithmetic, calculus, collaborative learning, mathematics curriculum

1. INTRODUCTION

A unique feature of effective computer-based learning environments as compared to other types of learning materials is their intrinsically cognitive character. Concrete material such as blocks for the learning of counting and early arithmetic, or mechanical drawing systems, or audio-visual technologies do not embody the key feature of a computer-based learning environment: it computes formal representations of mathematical objects and relationships. The interaction between a learner and a computer is based on a symbolic interpretation and computation of the learner input, and the feedback of the environment is provided in the proper register allowing its reading as a mathematical phenomenon.

This cognitive characteristic has bred lofty expectations in mathematics education on the assumption that computers will enable a deeper, more direct mathematical experience. This educational expectation is of a different nature from the expectation rooted in Computer Aided Instruction, because it involves changing the mathematical experience of learners at the epistemological level, rather than facilitating or automating a particular pedagogical style. But we need to judge the impact of computers, and whether they have achieved those lofty expectations, in two dimensions: (1) what the economists might refer to as "market penetration," and (2) epistemological penetration. Relative to (1), the expectations of thirty years ago have surely not been met. Computer-use remains a relatively small part of classroom practice, and the belief that computers will facilitate learning on their own has been seriously eroded. But relative to (2), the expectations of thirty years ago have been surpassed. Computers provide ways of doing and experiencing mathematics that we simply did not dream thirty years ago. The touchstone of these experiences is the experience of direct manipulation of mathematical objects and relations - a new experiential mathematical realism.

Many factors will affect the actual presence of technology in schools beyond the negative effects of the historic inertia of school practice and isolation of schools from the larger society. Among these are: merging of graphing calculators and low-end computers, the merging of communications and interactive computation technologies, the gradual rise in mathematics and other curriculum that requires a computational medium, the natural demographic effects of replacing school personnel who were educated before electronic technologies by people who grew up in a technologically rich world. While change takes time, and deep change takes even longer, we need to prepare and plan for coming developments, to understand better both the contributions and the limits of these technologies, and to have a clearer image of how practice may productively involve them. It is to this end that this chapter is directed.

First we rapidly survey the state of the art in the various domains of mathematics teaching, focusing where we think the source of deepest long-term change is occurring at the level of mathematical experience. Then we will consider the potential contributions of such emerging technologies as Artificial Intelligence (AI), and connectivity technologies in shaping productive applications of computers in learning and teaching within self-contained classrooms as well as at a distance.

2. THE NEW REALISM OF MATHEMATICS

2.1. MICROWORLDS TO EXPRESS MATHEMATICAL IDEAS

The first uses of computers in mathematics were to compute numbers, apply numerical techniques for solving equations, and check certain properties in analytical geometry. In the late 1960's symbolic manipulations, and then in the 1970's graphical representations, of functions became possible. By 1980, these had found their way into educational environments, and by the late 1980's bi-directional links between character string and coordinate graphical representations of functions became available as the first step in directly manipulable mathematical representations that provide direct dynamical feedback in the proper notational register. The first applications of computers were close to programming, and the complexity of the interface was such that learning the software itself was a significant first step. But in recent years a new level of realism of mathematics objects has been reached as the interface enables one to express mathematical ideas using a communication medium as close as possible to the usual mathematical language, and, importantly, as the interface provides feedback that can be read "directly" in terms of mathematical phenomena. The ability to access mathematical objects through various linked representations or points of view requires a tight relationship between the internal computational representation and the interface, which can no longer be considered as a mere superficial layer as it was in previous decades.

Mathematical microworlds by providing a dynamic semantics for a formal system (Thompson 1985) allow the learner to explore simultaneously the structure of the accessible objects, their relations and the representation that makes them accessible (Hoyles 1993, pp.1-4). A microworld consists of the following interrelated essential features:

(i) A set of primitive objects, elementary operations on these objects, and rules expressing the ways the operations can be performed and associated - which is the usual structure of a *formal system* in the mathematical sense.

(ii) A *domain of phenomenology* that relates objects and actions on the underlying objects to phenomena at the "surface of the screen." This domain of phenomenology determines the type of feedback the microworld produces as a consequence of user actions and decisions (Balacheff & Sutherland 1994).

Thus the possibility of turning complex operations or objects into new operations or objects available for further use is built into the definition of a microworld. From this perspective, we would say that the microworld can evolve as the learner's knowledge grows (Hoyles 1993, p.3).

This is a significant difference between microworlds and most traditional simulation systems, which consist of models whose parameter-values can be input by learners, who then can observe the effect of the chosen inputs. But the initial models do not evolve as the learner progress. Instead, some environments change the model in order to obtain this adaptive behavior (White & Fredericksen 1986). The criterion for the evaluation of such simulations is (a) how well they evoke a model of reference for some "reality," and (b) the quality of the visual feedback they provide. For a microworld to be considered as a simulation, it must satisfy epistemological criteria with respect to some reference model.

From this characterization of microworld, which emphasizes the formal system underlying the display of visible phenomena, questions can be raised in a systematic way which may organized their exploration or designed, such as the following:

- Given the specification of an object or a phenomenon expected at the surface of the screen, does there exist a construction within the underlying structure of the microworld that will produce it?

- Is it possible to give a "simple" comprehensive characterization of the phenomena that the microworld can produce? This is proposing an operative semantic of the microworld which goes beyond a mere user-manual.

- Given a domain of phenomenology, is it possible to construct a microworld that produces it?

Let us consider two examples: Logo and Cabri-geometry. Superficially, both microworlds seem to share a domain of phenomenology made up of drawings that could serve the learning of geometry. But if we go more deeply, we see big differences in their domains of phenomenology. In Logo, drawings are static. You can draw and re-draw them or modified versions of them, but only by modifying the Logo code that produces them. In Cabri-geometry, you can directly manipulate the drawings in a drag mode using "free points" at the surface of the screen. In Cabri-geometry intrinsic drawings are everything that can be drawn with ruler-and-compass, whereas in Logo you can draw any "arbitrary" set of points by enumerating them or by generating them with functions defined on number segments. So these two environments differ substantially, and these differences are significant at a cognitive level.

In both environments, learners need to decide whether they have drawn what they intended to draw. The criterion for this decision is based on perceptual evidence, but this evidence has a different basis in each environment. In Cabri-geometry a shape must keep its assigned properties and internal relations as any of its basic components is dragged around the screen. However, even such feedback may not be obvious to learners, although it may appear obvious to expert eyes (Bellemain & Capponi 1992, pp.76-77). In Logo the static character of the object offers a more fragile basis, unless the student coordinates the visual feedback with an evaluation of the symbolic description (Hillel & Kieran 1987).

Hence to characterize the microworld we cannot remain at the level of the formal description, nor at the level of the description of the screen specifics of its implementation. A description of the related domain of phenomenology and the type of feedback it allows is required since learning will be the result of the adaptation of the learner to their environment. In this adaptive process, the way the learner makes sense of the feedback and derives from this an understanding of their own activity shapes the meaning they construct of the mathematics involved.

2.2. EARLY NUMBERS AND ARITHMETIC

For younger children, since it is widely felt that physical rather than cybernetic materials are more appropriate, relatively little software has been developed targeting the learning of early number and arithmetic. The commercially available programs are aimed at teaching and automating computational skills in the form of the traditional algorithms, as with program Math Blaster, popular in the United States (Math Blaster 1993). Of more interest are those systems that are intended to help develop underlying conceptual operations, such as conceptual unit formation (and decomposition) and operating on higher order units. These operations are used to reorganize quantities for purposes of reasoning and computing more efficiently. In the past few years, some work has been directed towards the development of cybernetic manipulatives for whole numbers (Kaput et al. 1996; Thompson 1992), as well as for fractions (Ohlsson 1987; Steffe & Wiegel 1994). Most of these exploit the linking of (more or less) formal notations for the quantities involved with screen objects or collections of such. But, they still do not fit the dominant computational motif of school arithmetic. Rather than helping students become quicker and more accurate at executing computational algorithms, they slow down the curriculum and the student - adding a flexibility and depth of understanding that seem not to be valued as much as computational facility.

The development and wider scale deployment of software in the domain of arithmetic are tied to the reconceptualization of the number and quantity curriculum, a process that is not very advanced at this time. Virtually all work to date seems directed towards improving understanding of the decimal number system and variants on the traditional algorithms used to compute with it. More general quantitative reasoning, either for discrete quantities (Kaput & Thompson 1995) or arbitrary quantities (Thompson 1993), are still in early stages of study. Such more general reasoning is likely to require more ability to deal with arbitrary quantities - and the more arbitrary units and groupings - drawn from everyday experience rather than the idealized quantities of traditional manipulatives. Such would include the groupings associated with hierarchically packaging and trading things such as candies in different sized containers, or translating between arbitrary currencies, where much more flexible strategies are needed than simply grouping by tens. Traditional arithmetic becomes a special case in this broader domain.

Looking at numbers more from the point of view of relationships and structures, work with spreadsheets can be used to support pupils in the transition from arithmetic to algebra (Sutherland & Rojano 1993; Filloy & Rojano, this volume). This transition involves manipulating general relations, operating on the unknown, working with functions and inverse functions, and developing formal algebraic methods. Pupils from primary to secondary education can learn to use the spreadsheet language to solve mathematical problems and these experiences form a basis from which more traditional algebraic knowledge can be developed. In this sense spreadsheets provide access to the potential of the algebra language, thus removing one of the main barriers to learning algebra. In addition the computer frees pupils from the arithmetic activity of evaluating expressions, thus enabling them to focus on the structural and algebraic aspects of a problem.

2.3. ALGEBRA, CALCULUS, LINEAR ALGEBRA, AND THE FORMALISM PROBLEM

Most of the use of interactive electronic technology in algebra, calculus and linear algebra has centered on facilitating students' ability to use traditional formalisms and graphics: manipulating algebraic expressions, graphing functions, etc.. Computer Algebra Systems (CAS) such as Maple (1990), Mathematica (Wolfram 1995), MatLab (Moler 1995) or Derive (1994) enable students to define, combine, transform, compare, visualize and otherwise manipulate functions and relations in any of their traditional representational form[1]. As a result, traditional courses of instruction in these fields are undergoing significant changes, changes that are widely discussed in conferences and the literature (Hillel 1993). These changes reflect new, more exploratory approaches based on the interactivity, and more realistic problems can be addressed due to the computational support of work with "messier" symbolic expressions and wide-scope numerical solution methods rather than narrow-scope symbolic techniques (Fey 1989a; Fey 1989b). These also shift the balance of technique from deductive and algebraic toward inductive and empirical.

But the real tough didactical question is to discern "how much of the manipulative aspect can be eliminated while still sustaining conceptual learning" (Hillel 1993, p.29). CAS open a field of experimentation which calls for competencies more or less hidden in the classical curriculum, like what Hillel (ibid.) calls "window shopping" for results. What is seen on the screen, the graphical representation of the graph of a function, must not be trusted without being aware that what is involved is not mere perception but interpretation (Dreyfus 1993, p.125). For example, depending on the scale on the x- and y-axis, or of the sampling of the points used for the drawing, or the location of the window on the graph represented, the properties offered to perception may be quite different, ranging from true analytical properties to superficial visual effects. This unavoidable problem, because of the computational transposition of mathematical objects (cf. § 2.6), requires a careful choice of the situations offered to students (Artigue 1996), and a careful exploration of the screen phenomena likely to occur and potentially mislead students.

However, the curricular impact of these innovations has been local rather than global in the sense of changing what takes place in courses at the various levels, but not changing the larger relations among courses or the ways in which the big ideas of these courses appear in the mathematical experience of most students, particularly at the lower levels (Kaput & Nemirovsky 1995). Closely related to this fact is the highly intransigent "Formalism Problem," reflected in our inability to enable students to use their cognitive and linguistic powers to build meaningful connections between their everyday experience and the world of formal mathematics (Kaput 1994). In these terms, much of the technology referred to above has been directed towards facilitating movement on the formal island rather than moving off of it into students wider world of experience.

Whereas the last stage of technology use in algebra and calculus concentrated on manipulating and linking formalisms, the next stage likely involves mixing simulations and real data based in students' physical experience over a much longer age span. What remains to be explored is our ability to use technology to link personal and mathematical experience at a new level of intimacy. These sorts of activities do not preclude traditional activities involving movement on the formal island, and indeed, may increase the significance and effectiveness of those activities. Furthermore, students can switch among such motion simulations as an elevator, or characters walking in a scene, swimmers in a pool, drivable vehicles, or a more idealized, schematic and abstract simulation that represents all objects as points - all of which can be controlled by student-editable graphs, or in other ways (cf. §4.3 for an example). The students can get direct experience of the same model controlling more than one "world," and also get the experience of seeing models of intermediate abstractness (White 1993).

[1] While less flexible at the present time, and graphically much less powerful, graphing calculators are evolving to support much of the same activity (cf. Ruthven, this volume).

2.4. GEOMETRY

Geometry offers interesting recent developments based in the new access to direct manipulation of geometrical drawings, which enables us to view conceptualization in geometry as the study of the invariant properties of these “drawings” while dragging their components around the screen: the statement of a geometrical property now becomes the description of a geometrical phenomenon accessible to observation in these new *fields of experimentation* (Boero 1992; Laborde C. 1992a).

Logo first, at the beginning of the 70s, built a specific bridge between geometry and graphical phenomenon. But the user was still obliged to operate through a symbolic language which carried its own complexity. Logo is completely defined by a set of primitive actions and objects (i.e., numbers & lists), and a syntax that defines allowable combinations of actions and manipulations. It enables one to turn an organized system of primitive actions into a single complex one by combining them into a procedure that itself can act as a building block in other procedures (cf. §2.1), and its control structure is recursion, which means that a wide variety of self-similar curves can be constructed. And furthermore, its geometry is simultaneously differential (Abelson & diSessa 1981) and "from the turtle's point of view" (Papert 1980)

The Geometric Supposer (Schwartz and Yerushalmy, 1984) made a crucial step by offering the possibility of obtaining modifications of the current Euclidean construction without the necessity to restate completely its specifications.

But the accomplishment of the links between geometry and its experimental field, drawings of geometrical figures, that these environments looked for, has been reached by Cabri-geometry in the middle of the 80s (Laborde J.-M. 1985), and then the Geometers Sketchpad (Klotz & Jackiw 1988). These dynamical geometry environments are completely defined by a set of primitive objects (point, line, segment, etc.) and of elementary actions (draw a perpendicular line given a point and a line, a parallel line, etc.). It allows an organized set of primitive actions to be turned into a complex one using macro-constructions. The drawings produced at the surface of the screen can be manipulated by “grabbing” and “dragging” around any point having sufficient degree of freedom (Laborde C. 1993).

The status of drawings is changed by their availability in such environments, since they do not refer to one singular object but to a class of objects which share the same constraints and hence invariant features. Distinguishing between drawings and figures (Laborde C. 1993, p.49; Parzysz 1988) can then become a visible part of the geometrical activity of the learner: the drawing is obtained as the result of an explicit description of the objects and their relationships; conversely the observation of the behavior of the drawing in the direct manipulation allows the student to check the “correctness” of the intended figure, where “correctness” is experienced as the relation between expected invariants and observed invariants. Colette Laborde (1993, p.62) refers to this as the double role of visualization in geometry, a double role which remained hidden in paper and pencil environments because of the very low level of reliability of the production of drawings and the very small number of possible experiments. Beyond general common features of these learning environments remains the question of the kind of mathematics which can be learned as a result of a problem-solving activity in such contexts (cf. §2.1). In other words the actual geometry of Cabri-like environments is still an open question (Goldenberg & Cuoco 1996).

Considering differences between them is somewhat easier. Let us contrast Logo and Cabri-like software by considering the construction of a parallelogram. In Logo, a parallelogram must be described as a specific track of the turtle, which is defined in terms of relationships between angles and lengths of sides. Whereas, in dynamic geometry systems, it can be characterized either by the parallelism of opposite sides, or by the equality of their length, or by a property of its diagonal. This already suggests important differences, but there are even more since these intrinsic characterizations must be coordinated with characteristics of the microworlds themselves:

- In Logo the learner must take into account (explicitly or in action) relations between external angles of the parallelogram and internal angles because one is defining how far the turtle must move and how much it must turn (coordination of angles, directions and length).
- In dynamic geometry the learner must realize that given one point and two directions (parallelism criterion) or three points (equality criterion), the parallelogram is completely determined.

The understandings needed to make the construction are very different in these two systems, and so are the expected learning outcomes.

As noted by Chazan and Yerushalmy (1995, p.8): "teachers using geometry construction programs try to create experimental environments where collaborative learning and student exploration are encouraged." This very general movement in mathematics education is enhanced by friendly interfaces and direct manipulation. It raises the question of the nature of the tasks and problems likely to stimulate such an activity. In fact most of the classical problems used in classrooms become obsolete because of the efficiency of the Cabri-like environments, but at the same time they open the possibility for problems which were out of the scope of paper and pencil constructions - for example, to construct through a given point a parallel line to a given line using only point symmetry or reflection, or exploring the properties of the Poincaré model, etc. Among the didactic possibilities is to raise the plane of action and reasoning to a new level, where *sequences* of constructions are the objects of study, rather than individual constructions. In this case one can examine commonalities and differences across constructions because one is not forced to "live mathematically" within a given construction. This is analogous to what happens with families of functions in computer algebra systems.

Very recently the field of direct dynamic experimentation has been opened to new objects, the conics, where one can manipulate hyperbolas, elipses or parabolas or their equations. Almost nothing is known about the learning that could be expected, so this is also a new field of experimentation for mathematics education (Trgalova 1995).

Environments aiming at facilitating the learning of mathematical proof may as well benefit from these developments. This domain has been addressed mainly by researchers in Artificial Intelligence (AI), developing environments which reduce de facto proving to its more formal aspect. A prototypical example of this is the Anderson Geometry tutor (Anderson et al. 1985) which requires the student to construct the proof tree of some statement, providing immediat feedback on errors and limiting the learner to a linear top down or bottom up construction. It keeps the learner away from the problem-solving activity which is tentative, non linear, involving conjecturing as much as proving. The development of Cabri-like environment on the contrary contributed to support changes in teaching by stimulating conjecturing and the renewal of dialectical relationships between proofs and refutations (Chazan 1990). This significant contribution of technology to the evolution of school practice, indicates a direction for further evolutions of computer-based teaching environments in geometry, or other mathematical domains (cf. 3.2). Bridging proof environments and Cabri-like geometry environments, which mixes inductive/empirical reasoning and deductive reasoning, is another promising area to be developed in the near future.

2.5. MATHEMATICAL MODELING AND STATISTICS

It is common to recognize that “modern computing has revolutionized the practice of statistics” (Biehler 1993, p. 70). But from this revolution in the professional world to the revolution in the educational world is more than a small step. As Biehler (1993) clearly shows, to import professional software to the classroom is not sufficient and even causes specific difficulties.

In contrast to the obstacles to the use of professional statistics software in the classroom, there is clear evidence of the impact, actual and potential, of these environments on the teaching and learning of statistics. They allow one to deal with the quantitative complexity inherent to statistics, and they are necessary to approach data analysis in a way that allows one to explore the role played, respectively, by data and methods, and by chance and probability models. The tasks made available by these computer-based environments facilitate a shift of competence towards choice, combination and analysis of methods, and away from direct computation. It is obvious that statistics software allows one to bridge mathematics and “real” life by opening access to modeling of concrete situations and real data. It also offers a more general understanding of mathematics by the new kind of investigation of mathematical objects they allow and call for: e.g., comparison of functions by means of their graph, manipulation of matrices and vectors as tools for modeling (ibid. pp. 98-99).

The main current issue is then the design and implementation of statistical software offering a synthesis of the representation and computation functionalities of current professional tools on one hand, and facilities to allow the kind of investigation needed for the learning and understanding of exploratory analysis (allowing the variation of data as well as of methods) on the other.

TableTop (Hancock 1995; Hancock et al. 1992) offers an example of deepening impact of technology on the relations between mathematical and non-mathematical experience. In this software, each item in the database is represented by an object on the screen, so that one can create databases involving the students themselves (or their toys, or pets, etc.), and then, as one sets up, say, a scatterplot, the objects move to their appropriate place on the screen. Students can control the visual nature of the screen objects, and the objects become closely identified with their creators. This kind of personalizable, live system is much easier for students to use and to relate to than the highly abstract and idealized row/column representation that has traditionally been used to represent data. Problems of concreteness and personalization come to the fore as students attempt to abstract and idealize relationships in data. As with any educational environment, however, the key ingredients are the actual activities in which the students and teachers are involved (Hancock et al. 1992).

Biehler (ibid. p.96) mentions additional general features which would be helpful for didactical purposes:

- Cabri-like functionalities to move data points by mere dragging of points, in order to facilitate the experimental study of the dependence on individual points;

- Graphical input of data for “sketching” data sets with required geometrical properties;

- Manipulable lines and shapes for defining experiments[2];

- Allowing dynamic annotation of representations, like markers of the mean or median of an histogram.

---

[2] As in Statistics Workshop (Rubin & Bruce, 1991), where the user can change a line in a scatterplot by pushing it with a mouse and observe the numerical effect or the qualitative visual modification.

2.6. COMPUTATIONAL TRANSPOSITION AND RELATED ISSUES

In the previous examples we examined the contribution of the computer-based learning environment to students' mathematical experience and how differences in design and implementation affect that experience. The actual implementation of a learning environment requires decisions at the programming level in order to cope with the constraints of the operating system of the machine, the specificities of the programming language and of the related representations whether they are internal to the machine or at its interface. In particular, mathematics is directly concerned with the internal representation of numbers, the sampling of data required to represent a function, and display a representation of its graph at the interface (which has as a direct consequence the window effect mentioned by Hillel - cf. §2.3). Data structure and choice in knowledge representation play a crucial role as well.

Consider the two environments, PIXIE (Sleeman 1982) and APLUSIX (Nicaud 1992), addressing issues related to the learning of elementary algebra. Such algebraic expressions can be represented as strings of characters, e.g., 5x+2x(x-3), or as tree structures. But the choice of representation system determines the kind of manipulation possible at the interface of the system. If the tree structure is chosen, as is the case of APLUSIX, then some manipulations are not possible. For example, 5x+2x cannot be extracted from 5x+2x(x-3) and the transformation 5+3x -> 8x has no interpretation. On the other hand, the choice of a list structure, as is the case for PIXIE, allows such manipulations and thus the possibility of successes and failures of a learner at the level of basic algebraic manipulations. Actually, these different choices in design must be understood considering the different teaching objectives of these software: the former addresses *strategies* for factorization of elementary algebraic expressions, the latter addresses *manipulation of algebraic expressions* related to solving equations of degree 1.

So, knowledge is transformed in the process of implementing educational software because of computational constraints as much as it is transformed under the didactical constraints. Necessary decisions to be taken for the design of a software, such as the choice of a knowledge structure and representation, or of the algorithms to apply, or of a grain for the description of the objects, imply a *computational transposition* (Balacheff 1996) which consequences on knowledge are as crucial as the already known didactical transposition (Chevallard 1985; Kang & Kilpatrick 1992).

Direct manipulation of objects and reification of knowledge, together allow experimentation in domains that previously required a high level of technical skill or background knowledge. But, since students' knowledge is an emergent property of a dialectic relationship between perception and conceptualization during interaction at the system's interface, which in turn depends on the structure of the internal computer formalisms, the choice of phenomenology that these are linked to, and details of the interface implementation ultimately determine the educational potential of the technology. For this reason, specification, implementation and didactical evaluation of educational technology requires the analysis of the way knowledge is defined, represented and finally implemented in systems. As a result, in some cases analytic and visualization tools may need to be specially constructed for learners according to principles different from those used to construct experts' tools (cf. for example the Envisioning Machine software (Roschelle 1994).

But, despite the fact that these decisions about teaching or about student experiences are often based on implicit or ill-defined information, designers and developers nonetheless participate as well in a didactical transposition of knowledge (Balacheff 1996). At the intersection of didactical transposition and computational transposition stands the problem of the relationships of the knowledge taught as it results from the behavior of the system with the knowledge its designer intends it to teach. What is the experienced epistemological content of the software? Might there be new mathematics as a result of its computational embodiment?

Not only are the domains of mathematics changing, but the relations between them are changing as well. For example, once it was assumed that you needed to know algebraic symbol manipulation in order to do the symbolic manipulation involved in calculus. Early application of technology was to use algebraic symbol manipulators in the doing of calculus. Then, as described above, students encounter the underlying ideas of calculus in phenomenological and graphical terms before any algebraic work, and the algebraic descriptions of phenomena arise in a support role, often "overlaid" on the prior experience.

The order in which actions take place could become arbitrary in the eyes of users, which can have significant consequences. Consider an example in geometry: an equilateral triangle is completely defined once one of its sides is known. So it can be defined for Cabri-geometry as a macro-construction following the classical construction using two circles of the same radius |AB| centered on the extremities of a given segment AB. The arguments of this macro-construction will be the extremities of a segment. But if the user gives the argument in the order A then B, or B then A, the triangle drawn has a different orientation with respect to the segment AB. This demonstrates the impact of the orientation of the plan which is in general forgotten in elementary geometry, but is recalled to the user as a result of the sequencing of actions (Payan 1992).

Reacting to what might be seen as technological limitations; one might suggest other implementations to eliminate the effects. But new implementations would give rise to new "side effects," or unintended ones. The question is not to suppress them, since any interactive representation has substantive effects, but to be able to specify in detail what they are. Multiple representations are often suggested as a possible solution for this kind of problem, but it supposes that one can exhaustively enumerate and describe representations related to a given piece of knowledge. A quick look at the history of human knowledge, even restricted to science, or at the various representations punctuating the development of learners, makes clear that the possibilities for productive notations are unlimited. Another solution which has not yet been adequately explored is to characterize the domain of validity of the chosen representations, and thus the domain of validity of the educational software itself.

These problems and phenomena are intrinsic to computerization of mathematics and learning environments. Adapting and representing knowledge within the computer medium has inevitable effects, which constitute both constraints and opportunities. Dealing with these constraints and opportunities involves design issues beyond education - in knowledge engineering, computer graphics, person/machine interaction, for example.

3. MODELLING TEACHING AND LEARNING IN MATHEMATICS

3.1. AI AND MATHEMATICS EDUCATION

The first significant projects of A.I. in the field of mathematics education appeared in the early seventies, for example, the Integration Tutor from Kimball (1973). Projects of this period tended to be A.I. projects taking Education as a field of application rather than projects in the domain of Educational Technology. At the same period the LOGO project started (Papert 1973), one of the first significant A.I. efforts with application to Mathematics Education (Hoyles 1993, pp.14). In the eighties we see many A.I. projects specific to mathematics education, for example: BUGGY (Brown & Burton 1978), PIXIE (Sleeman 1982), ALGEBRALAND (Brown 1983), GEOMETRY-tutor (Anderson et al. 1985), WEST (Burton & Brown 1979), etc. In the early nineties the number of A.I. projects directed specifically to Education grew further.

Because of an initial focus on the design of autonomous machines, AI applications showed an almost universal lack of attention to the role of the teacher. Another feature of traditional AI approaches are two separations: (1) between "content" and heuristics, and (2) between control/inference structures and the content-related interface. In mathematics education, not only is the teacher indispensable, but these distinctions are likewise problematic. A combination of mathematics' essential use of multiple notation systems, its inherent epistemological layeredness and complexity, and the embedded subtlety and implicitness in the role of heuristics, doom from the start any large scale attempt to automate teaching. Another barrier is the manifold complexity and fluidity of the mind, intention and learning. And finally, we have the subtlety, implicitness, and distributed nature of the didactic contract which emerges as a result of the teacher and learners negotiation of the meaning of any situation.

Given the challenges in each of these interacting dimensions, the relationships between mathematics education and AI, as is the case more generally for all technology (Kaput 1992, p. 516), gives new life to virtually every traditional didactical issue.

The design and implementation of computer-based learning environments, beyond the initial dream of an autonomous artificial intelligence, are extremely complex. Because of its intrinsic cognitive nature, it forces key questions related to computational and cognitive modeling which are at the core of AI in a modern technological sense.

3.2. MICROWORLDS VERSUS TUTORING SYSTEMS

Microworlds and tutoring systems have stood at two extreme points of a continuum of computer-based learning environments organized on the basis of their didactic directiveness. On the one hand, microworlds offer to learners open worlds in which they can freely explore problem situations, and on the other hand, tutoring systems provide students with strong guiding feedback. But in both cases merely interacting with the machine is insufficient.

Microworlds' free exploration offers a rich range of experiences but does not guaranty that specific learning will occur: "As far as exploratory software is concerned one side is the facilitation of student-generated pathways to mathematical knowledge: but simultaneously, the other is the inevitability that students might not accept or even notice the educator's agenda" (Hoyles 1995, p.206). The student may focus on screen events not relevant to mathematics learning. So how to overcome such difficulties?. One part of the answer is to understand better how learning takes place in such contexts, which requires new research, including new forms of research (Hoyles 1995, p.217). Another part of the answer lies in the search for design principles for teaching situations and teacher management involving microworlds, where such characteristics could ensure an expected learning outcome (Hoyles 1993; Chazan & Houde 1989).

In the case of tutoring systems, the close interaction of a tutor can guarantee certain performances but does not determine the nature of the underlying meaning. One reason is that the learner usually cannot express his or her view about the knowledge concerned, so the tutor feedback focuses more on the knowledge of reference than on the learner's knowledge and its evolution. Another reason is that learning in such an environment could mean learning how to obtain the best hints and help from the tutor so that the problem at hand can be solved. In other words, the student can learn how to optimize the use of the tutor feedback instead of the knowledge the task is supposed to convey (e.g.: Du Boulay 1978).

The Geometry Tutor (Anderson et al. 1985; cf. also § 2.4) is a prototypical example of an intelligent tutoring system. It guides learners in the construction of a mathematical proof in geometry, providing immediate feedback, clear hints and help when the learner fails or gets lost, but it accepts only those learners' explorations that are likely to lead to a correct proof (Anderson et al. 1990, p.18; Guin 1991). Some tutors impose less restriction on learners. For example, MENTONIEZH (Py & Nicolas 1990) supports quite flexible investigations of mathematical proof, but the construction of the proof while not necessarily "linear", must still be on the correct track.

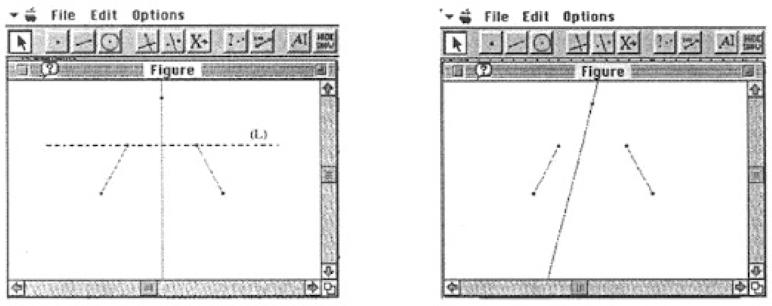
A trend of current research is to look for environments with a better balance between these two extreme points. Guided discovery software (Elsom-Cook 1990) suggests to adapt the level of directiveness to the current state of the learners' knowledge, varying from very open, microworld-like environments for some purposes, to tutor-like behavior if the learner seems to need support, or if the teaching target appears to be better reached in this way. In such environments the more difficult task is the evaluation of the learner's activity in the microworld.

In the case of geometry a crucial question is the extent to which the drawing produced is generic (Laborde C. 1992b) and correctly fulfills the initial problem specification (Allen, Desmoulins & Trilling 1992). Guided Discovery Learning Environments offer the promise of educational software closer to a constructivist conception of the construction of meaning, and escape from the programmed teaching and behavioral learning paradigm. Solving these didactical problems may require generating scenarios facilitating the engagement of learners in relevant situations (Hoyles 1993; Vivet 1992).

3.3. UNDERSTANDING LEARNERS' UNDERSTANDING AND INTENTIONS

When the machine intends to teach, it has to interpret the student's actions performed at its interface in order to generate adequate feedback. When these actions do not produce events expected by the designer, an interpretation is needed in order to decide whether the gap between the expected and the actual events is significant. Such gaps could range from mere interface slips to errors as symptoms of underlying misconceptions. This difficult problem is referred to by the AI community as *student modeling* and represents an ambitious research program. Several lines of research have been explored, from the implementation of an *a priori* catalog of errors to the use of machine learning algorithms, the machine attempting to induce automatically mal-rules which might "explain" the observed gaps. The student modeling problem is made even more difficult by the noisy information at the interface (Chan 1993). For a detailed overview of these approaches, see Wenger (1987, esp. pp. 345-394), and, for a more advanced state of the art presentation see McCalla (1994)

The interface-noise problem becomes more difficult with environments allowing direct manipulation of graphical objects at the interface (Laborde C. 1993). Consider the construction of the image of a line-segment by a reflection of a given axis, using Cabri-géomètre. The figure (Fig. 1a) shows a screen image of the solution of a learner A*. This drawing seems satisfactory, but when one moves the axis of symmetry (Fig. 1b) it appears that it no longer fulfills the constraints of symmetry. Actually, A* has not constructed the auxiliary line (L) as a perpendicular line to the axis, but as a line perceptually horizontal (the pixels are aligned) while the axis is vertical. For Cabri-géomètre, (L) is a line with no specific constraints, so when one of the basic points is dragged, the perpendicularity is not maintained.



(a) (b)

Figure 1.

Actually these errors, and similar ones reported by Bellemain and Capponi (1992), could be a mixture of true geometrical errors and errors related to the student's understanding of the behaviors of the learning environment itself (Hoyles 1995, p. 208)

Although they do not explicitly implement models of the users, microworlds likewise must confront the issue of users' intentions. In the above example, should the environment have identified the perceptual perpendicularity of the lines and ask the learner: "would you like these two lines perpendicular?" This suggestion could appear too extreme to a teacher. But the same problem occurs when a point is constructed almost on a line. Should we interpret it as a point on the line or should we request the user to express it before hand? The answer requires a difficult design decision involving tradeoffs made even more difficult because the student's intentions may change drastically as he or she learns.

Student modeling offers great expectations since it promises systems with more relevant behavior. But, it raises tough questions for both researchers in mathematics education and researchers in A.I. Interestingly, the failures in this domain result as much from our lack of knowledge of learner mathematical conceptions as they result from A.I. itself. More systematic research on student conceptions is needed to realize the dream of student modeling.

3.4. LIMITS OF AN AUTONOMOUS MACHINE

The shortcomings we have emphasized, including those of Guided Discovery Learning, might not disappear that easily. Since these systems have teaching as an explicit objective, they cannot escape the emergence of a didactical contract and its related consequences. So learners could still construct knowledge in order to satisfy some machine expectation instead of a construction specific to the intended problem-situation, or construct misunderstandings. The didactical problem then is not to avoid or to make disappear such learning phenomena, but to understand their origin and specify conditions needed for evolution of appropriate conceptions. Many errors are due to conceptions whose existence is necessary for the construction of meaning (Confrey 1990). Learners must both construct these conceptions and then overcome them to have access to the intended meaning (Brousseau 1986). This problem is very difficult because these conceptions can evolve in unintended directions so one cannot know a priori if the overcoming of a difficulty or obstacle amounts to progress (Balacheff 1991).

These questions expose the difficulty we encounter in considering learning environments as autonomous systems. To overcome such difficulties we must consider the computer-based environment not as an isolated system, which was the initial position of A.I., but as part of a larger system which includes the teacher. We will consider this system from two points of view: the machine as part of the didactical situations organized by teachers, or the machine as a partner of the teacher.

4. COMPUTERS IN THE FIELD

4.1. TEACHER/MACHINE PARTNERSHIPS, WHERE ARE THE PROBLEMS

Teachers cannot exploit new technology in their daily practice if they are not well informed on its place and role in a didactical process. We might say that they must know the computer based system from a didactical point of view in the same way that they would like to know a colleague with whom they might share the responsibility of the class. Study of these situations and their functioning are at their very beginning. What knowledge should be "encapsulated" in the machine and how should it function relative to a given learning/teaching target? And how should teacher and machine communicate? Much AI research has dealt with explanation for teaching purposes, all centered on learners, none on teachers and the explanations they might expect from the machine about a didactical interaction. The challenge is providing an appropriate level of relevant didactical information in order to support the teacher. This is an open problem for researchers in both mathematics education and computer scientists. Meeting it is one of the conditions for tomorrow's cohabitation of teachers and machines (Balacheff 1993; Cornu 1992).

The necessity of tightly defining intentional knowledge is often raised as questionable. For example, in the early days of the Logo Math Project (Hoyles & Sutherland 1989), students were allowed to construct their own problems, projects or goals. No intentional knowledge, in academic terms, influenced the situations organized. Some pupils learned some mathematical ideas, but haphazardly, depending very much on the projects pupils happened to create. While Logo Math Project students were expected to learn some algebra-related ideas, they did not spontaneously generate the need for the idea of a variable in Logo. Hence an intervention from the teacher was needed in order to "negotiate" with learners the meaning of a situation which was likely to lead them to some specific learning. With such interventions, students did learn algebra-related ideas (Sutherland 1993).

Negotiation does not refer here to an explicit process resulting in an explicit agreement. More often than not this agreement is the emergent outcome of a series of interactions between learners and the teacher gradually leading the learners to accept a certain task, activity or problem as their own. Such interactions occur under constraints of time and subject-matter responsibilities (Arsac et al. 1992), as well as cognitive constraints, classroom culture constraints, and the social need to share meaning and communicate with the world outside school. The intrinsic complexity of this negotiation arises from the need to leave enough autonomy to learners, without which there is no meaningful construction of knowledge, yet "guarantee" that teacher's intentions will be fulfilled (Brousseau 1992).

As Hoyles and Noss (1992, pp. 32-33) point out, the literature is quite sparse in outlining specificities of pedagogies that balance the learner's initiative and the teacher's intentions. This applies to the literature on microworlds as well to more general computer-based learning environments. Furthermore, teachers must be aware that the didactically central features of the environment may not be obvious to students. For example, as reported by Bellemain and Capponi (1992, p.83) in the case of Cabri-geometry, not all students spontaneously consider robustness of drawings under dragging of free elements as intrinsic to the validity of a construction; it was necessary for the teacher to engage in a specific negotiation in order to give clear status to this feature. Also, the learning situation may extend beyond the obvious direct learner/machine interaction. For example, in the Logo construction of a parallelogram (cf. § 2.4), an activity away from the computer is likely to help the student make sense of the role of the symbolic representations and their relation to visual effects (Hoyles & Noss 1992, pp.137 & 165-166). Here, the learning situation includes the period away from the computer as well as the subsequent use of the microworld.

In order to adapt computer-based environments to the specifics of their actual classrooms and their own objectives, teachers must be able to customize them. This is classically the case in Logo, within which many ad hoc microworlds are constructed, e.g., a microworld for geometrical transformations (Edwards 1989). This is also the case in Cabri-like environments by suppressing some menu items or adding macro-constructions as new menu items, as in the learning environment for symmetry developed by Capponi (1993).

Another important issue is how microworlds allow teachers to control of the learning process, its outcomes and the meaning learners are likely to construct. This requirement might appear too ambitious to the eyes of some teachers or some researchers, who point out the very often unpredicted, and possibly unpredictable, variety of learners' procedures and strategies (Hoyles & Noss 1992, p. 138). A way to manage this complexity is to explore the sensitivity of learners' problem-solving to modifications of the problem situation, and to consider as variables of the situation those characteristics whose modification is likely to influence learners' strategies (Artigue 1992, p. 50). Activity away from the computer can be such a variable, since it may play an important role in shifting learners from doing to thinking about their action. Other variables more specifically related to the problems being considered could play a similar role, as for example in the case of line-symmetry, the respective position of the axis and the object. To identify those variables for a given problem that are available within one environment but not another, or that are made available with different characteristics by one or the other, is one way to specify the didactical validity of a computer environment (Sutherland & Balacheff, 1996).

4.2. DISTANCE TEACHING AND THE VIRTUAL CLASSROOM

The recent integration of communication technology and computers gives rise to new expectations for distance education (Kaye 1992). Two domains currently being explored are (1) telepresence and virtual classrooms, and (2) customized teaching intervention at a distance.

In virtual classrooms, a teacher located at one site gives presentations to students located at geographically distributed sites. Two different implementations occur, one in which several students at each site can see the teacher on a video and possibly interact, and where the teacher can see the remote sites on several control screens; the other implementation consists of broadcasting presentations to individual students across different sites whose receiving machine may support interaction with the instructor, but where the teacher cannot see all the students at once on his/her machine, but may receive queries, to be answered either immediately or to individuals later.

The case of customized teaching interventions at a distance is somewhat different. In this case the teacher interacts with individual students in real time, perhaps assisted by AI-based support. He or she must, almost on the fly, diagnose the situation and decide what action to take. Such environments are often enhanced by a video channel allowing telepresence. For example, in the case of TéléCabri (Fig. 2) when the communication is established between the student and the teacher, the latter sees the current screen of Cabri-geometry and must make sense of what the student has constructed and why. A two-way messaging text window may provide more information about the drawing and its construction and perhaps some aspects of what the student has previously tried.

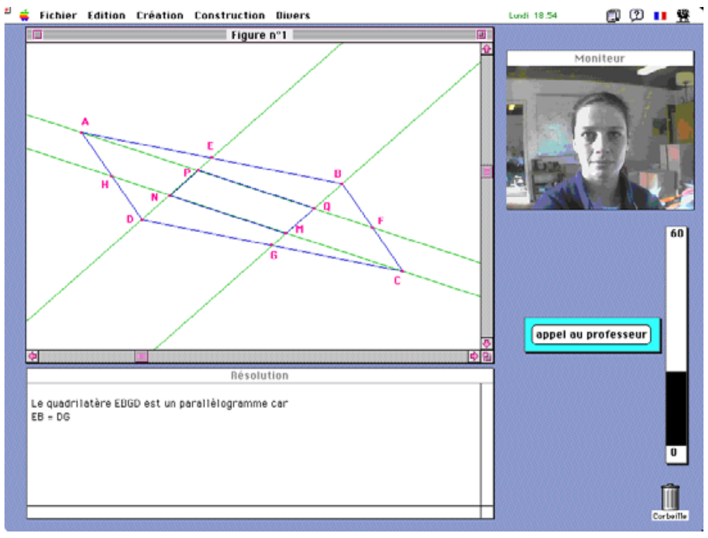


Figure 2.

Unlike a traditional classroom, the teacher cannot easily avoid this particular student's situation, the mathematical content, and the student's current understanding. In particular, a substantial dialog and a negotiation may be necessary. Both the student and the teacher must develop new behaviors and new views on teaching and learning. Because both the cost of the communication and the cost of the customized intervention of the teacher are high, the advantage of AI assistance is amplified, where the machine may contribute to the teacher' work, handling routine difficulties, for example.

These new forms of education will require new skills on the part of the teacher, skills largely unresearched from the point of view of the professional knowledge about the contextualization of mathematics by technology, as well as from the cognitive point of view. And likewise, research to inform decisions on how to distribute the intervention workload between the teacher and machine is needed more now than ever before.

4.3. DISTRIBUTED SYSTEMS AND COLLABORATIVE LEARNING, THE PARADE SCENARIO

Radical increases in connectivity, including interactivity at a distance, computer mediated conferencing, World Wide Web (WWW), etc., is likely to change the social structures of the education system, as well as access to data and expertise outside schools (Solomon & Roberts, in press). Distributed and virtual structures will influence centrally organized structures, introducing important modifications in them. Changes in connectivity also involve smoother mixing of "real" data and computer generated data (as generated by simulations, for example). These issues are too complex for a full analytical discussion within the space available, so instead we will offer a concrete and suggestive scenario, "Parade" discussed in more detail in (Kaput, in press).

The "Parade" scenario illustrates the use of a combination of the Internet as a platform and independently produced interoperable software components in ways that challenge many of our tacit assumptions about the how mathematics can be experienced by students in schools, how students relate both to teachers and to one another, and how teachers relate to one another and to instructional resources, including assessment resources. It also hints at the new mathematical content and types of modeling likely to dominate in the coming decades.

"Parade" is a team game played within a vehicle-simulation system, e.g., MathCars (Kaput 1993)

> In what appears to be a kind of slow-motion line-dance, two blindfolded 8th graders in Boston are following a leader in an irregular motion along the wall in the back of the room. The first student calls out his distance from the start in paces "1, 2, 3, 3, 3 ,4, 6, 8" The dance ends when the third student bumps into the second. They are rehearsing in front of a motion detector and video camera for an "instrument driving rule" version of Parade. According to instrument driving rules, each car only knows either the position or velocity of the car immediately in front of it, and its own position or velocity, but cannot actually see any other cars. The cars are to follow the lead car in a parade without crashing or getting "lost," which means falling so far behind that you lose contact with the car in front of you. A warning beeper sounds if you get too close to or too far from the car in front of you.
>
> The 8th graders will soon be sent a challenge trip from another team in Grenoble that determines the motion of the lead parade car that they are to follow. They are to drive their simulated cars behind the Grenoblois car. Each car behind the lead car has two graphs on its dashboard: its own velocity vs time graph as it is being generated, and the position vs time graph of the car immediately in front. Hence the act of staying in the parade requires coordinating velocity and position data in real time. The team from Grenoble has successfully completed the trip with a parade of four cars behind the lead car using position-velocity information. The Boston class is rehearsing with their own body movements to get a feel for the activity before ultimately trying a parade in the simulation with five cars.

A Classroom Scene

It turns out that one student has heard a rumor that the Grenoble group has actually used a person's motion in front of a motion detector to define the motion of the lead car for the challenge. Once the data has been imported into the computer, it can be "attached" to a simulated vehicle, which can re-enact the motion. Or the motion (its velocity graph, say), can be graphically edited and then replayed. The student is thus trying to create as irregular and unpredictable a motion as he can in the hopes of using it as the motion for his team's lead car if they match the Grenoble challenge (then they will send their motion to the Grenoble team as the next level in the challenge).

As the three Boston students carefully review the respective position and velocity graphs of the lead student obtained from the motion detector, they become intrigued with what appears to be *an exaggeration of velocity changes* with each following person and turn to the video to check if what they "felt" was actually the case. They marked all three students on the video, and then, plotting the velocity of each person, they realized that each successive follower seemed to have greater velocity swings.

The question arose: Why does this happen? Does it have something to do with reaction times, or is there something else involved? They decided on a simplified experiment involving the Mouse-Based Lab. Here, the lead student dragged a car back and forth on the screen to create a fairly simple wiggly, up-and-down position vs. time graph. Then the second student attempted to match that position graph by dragging the car on the screen with real-time-only feedback only on the velocity of the dragged car. Each of five students did likewise, but based on the position graph of the previous student's car. Then they plotted all the velocity graphs on the same axes. It seemed that there was indeed some kind of "exaggeration effect" going on, with whatever trend started by the second student becoming exaggerated in subsequent students.

Next they decided to ask others across the world whom they knew from previous interchanges to try the same experiment. Accordingly, they created another wiggly position graph by deforming a sine function, and, using voice annotation, described the experiment they wanted others to try. They then sent their notebook to their colleagues with a request to send back the series of their velocity graphs in their experiment reports. They also posted their experiment on their WWW server in the expectation of getting more data.

A group at one school decided to modify the experiment to see what would happen if the person driving knew in advance the whole position graph of the car in front rather than seeing it unfold in real time as the trip progressed. They claimed that there was no exaggeration effect, but another group found a data analysis system that revealed a small exaggeration effect in that same data. Meanwhile, another group of students decided to create a *dynamic system model* on differentials of the Parade motion with feedback rules based on separation distance: slow up if the distance falls below a certain threshold, speed up if it exceeds a certain threshold. This had the effect of simplifying things even further. Surprisingly, once again the exaggeration effect seemed to emerge in these models also - without people intervening! By adding fifty cars to their model, they noticed that a slight slow-down in the lead car could result in stoppage of the more distant following cars. Their teacher said that they had "reinvented the traffic jam." When they posted their result on the network, a group from the MIT Media Lab offered a new way to model such phenomena that did not even require a lead car. The new model even applied to geese flying in formation, where there is no lead goose, but only individual geese each following simple energy conservation rules. Indeed, the new way of modeling applied to a whole range of situations in biology and even sociology (Resnick 1994). Students in schools affiliated with this research center were anxious to share their insights and models with inquiring students involved in the Parade investigations.

4.4. THE DEEPENING IMPACT OF TECHNOLOGY ON MATHEMATICS EDUCATION: THE CONTENT AND CURRICULUM DIMENSION

We identify two stages in the deepening impact of technology on content and curriculum. The first stage is based on facilitating the handling of traditional representations and actively linking these. Such changes are accounted for in Fey (1984) and Schwartz & Yerushalmy (1993), for example. The second stage was discussed in our earlier sections describing technology's impact in various topic areas of mathematics. There we identified two ways in which the impact of technology is being deepened: (1) the direct manipulation of mathematical objects and relations; and (2) the linking of real experience with mathematical formalisms using combinations of simulations and real data (cf. § 4.3).

We saw (1) occurring in many domains, and illustrated it mainly in the context of "geometry" we use quotation marks to emphasize that some of what the dynamical geometry systems provide amount to new mathematics. It is no longer geometry in the traditional sense, particularly as one is able to program sequences of constructions and variations in parameters into a given configuration and then "run" the system to create dynamic animations. These result in a new collection of phenomena to be studied, phenomena and techniques that seem to cross between geometry and topology, with a new importance to questions of continuity (Goldenberg & Cuoco 1996; Laborde J.-M. 1996).

We illustrated (2) in the contexts of algebra and calculus. The Parade scenario exemplified a content which is not currently represented in most today's schools. The dynamical systems material, while introduced theoretically early in this century, did not flower until the availability of the computer medium in which rapid iterative computations are possible, which began in the 1970's and has been accelerating since (Gleick, 1987). Much of this material has graphical form that does not initially depend on algebraic language (Devaney 1992; Hofstadter 1981), and can arise from "naturally" occurring situations as suggested by the scenario and the work of Sandefur (1990; 1992; 1993). Such systems in the computational medium provide entirely new styles of modeling natural phenomena that are based on rules defined locally on the primitive elements of a larger system rather than modeling the global behavior of the larger system: one describes the fox/rabbit predator/prey system in terms of rules that describe how individual foxes and rabbits behave (Resnick 1994). From these locally defined rules, global behavior emerges. The mathematics of these models will become centrally important in curricula of the next century and are likely to constitute the next stage in the deepening impact of technology on mathematics education.

Such approaches to the learning of serious ideas does not necessarily need to be incremental students may occasionally plunge into ideas that have aspects that they are not be able to appreciate until much farther in their schooling, yet can make some sense of (Nemirovsky 1992). Similar approaches should make sense for the mathematics of uncertainty, data, space, and dimension, for example (Steen 1990). This computationally intensive mathematics (which simply could not exist apart from the computational medium), raises the importance of certain ideas in traditional mathematics, such as point-set topology, while other mathematical ideas such as dimension are generalized in new directions (fractional dimensions, for example). Thus, we suggest, the process of technology's impact on mathematics content and curriculum is continuing to deepen by changing the nature of the subject matter and how it is used to understand the world. The earlier processes that we have identified - facilitating the manipulation and linking of traditional notations, direct manipulation, hence reification of mathematical objects and relations, the linking of wider experience with formal mathematics via models and simulations- will not disappear, but will be included in the new changes as they occur, as new mathematics and new notations develop. We expect that the same can be said for the inclusion of the didactic changes that we have previously described. Provided that we have learned from the recent history of Logo, which "shows how easily a potential curriculum strand can be converted into a topic, made manageable within the margins of the conditions of schooling, and turned into a piece of the canonical curriculum" (Kilpatrick & Davis 1993, p. 208), we may be in the early stages of a transformation that will continue and likely accelerate into the next century, challenging curricula and accepted forms of educational practice.

But not only is the substance of the curriculum changing, its global organization is changing as well. Access to important mathematical ideas can be broadened to include a wider sample of the student population, and these ideas can be introduced to students at a much earlier age than traditionally - by carefully exploiting the concreteness and direct manipulability of computer representations, the ability to factor out complex calculations as needed, the ability to create links with everyday experience, and the ability to provide varying levels of didactic support. Hence traditional prerequisite structures based on computational competence are directly challenged, as are assumptions regarding which students can learn what mathematics. New degrees of freedom are available in curriculum and didactics design. Of course, the question of when and how these degrees of freedom may be exercised is not entirely a matter of the technology itself, and will vary widely across, and probably even within, different countries depending on political, cultural and economic factors beyond the scope of this paper. Such changes not only involve schools and schooling, but the relation between schools and the wider society, and the relation between education and schools, particularly as the connectivity revolution and the associated changes in technological infrastructure change the economics of education delivery. In the US, for example, the trend for education to “leak out” of the schools into homes and workplaces is well underway.

It has become increasingly apparent that the mathematics and science of the 21^{st} century will be more different from today's than today's is different from that of the 18^{th} and 19^{th} (Casti 1992; Kauffman 1993), especially since they now grow in a new, dynamic interactive medium and are becoming ever more intimately tied to economic and social destiny. As currently constituted in (mostly university level) curricula, this mathematics (and hence the associated science) appear at the end of a long series of algebraically oriented prerequisites that the great majority of students do not complete. It is available in formal language to those who master the formalisms (Kaput 1994). But across the world, we have done a relatively poor job of making those formalisms available to the great majority of students. The net result is that relatively few students reach the mathematics and science that currently depend on facility with algebra, and those who do, often do not learn the subsequent material especially well.

5. CONCLUSION

Over the last decade, we have seen enormous progress in computer-based technologies for mathematics education relative computational power of machines and software, friendliness of interfaces, and efficiency of connectivity. Promising new development are on the way, for example, in the combination of scripting and distributed components that "plug and play" despite being produced by independent organizations (Roschelle & Kaput, 1996). In particular, by using modular components whose configurations, interface and connections (including connections over the Internet) can be controlled externally via scripts (e.g., AppleScript, Visual BASIC, Java, etc.), the larger educational technology enterprise is able to do collectively what no individual organization could do on its own. A critical feature of scriptable systems is that the huge complexity of didactical interactions can be managed by focusing on one activity or exploration at a time. Furthermore, since the investment in scripting is quite modest at each stage, one has the flexibility to make changes either to improve the activity or to adjust it for different audiences[3].

[3] A substantial series of activities centered on the mathematics of change is being developed in this way as part of the SimCalc Project (Kaput, 1995).

Despite all this progress and promise, the penetration of these technologies in educational practice proves to be very slow and with great disparity from place to place. The actual availability of machines or access to the net is surely one reason. But one can easily imagine that in a not-too-distant future, fully networkable palm-top computers running educational software components will become as common and as cheap as today's graphing calculators[4]. Students will then have the full power of this technology on hand at all times, and schools will be able to overcome the current technology shortage. But the commitment of teachers and our large ignorance about teaching and learning in these contexts are at least as important as factors to overcome the discrepancy between technological availability and actual use in classrooms.

Obviously, it is not sufficient to provide learners with carefully designed, powerful software, with nice graphical representations and friendly interaction, to ensure that significant learning will occur, or to identify what kind of learning it is. For teachers, traditional professional knowledge is not sufficient to deal with the deep changes in learning, teaching, and epistemological phenomena that are emerging. The depth of this change, complexifying the teaching/learning situation, is a result of the fact that the new systems are primarily materializations of a symbolic technology, the means by which mathematics is knowable. There are two dimensions in which the computer makes its primary impact: (1) symbolic, by changing the representational medium in which mathematics is expressed; and (2) interactivity, by changing the relationships between learners and the subject matter and between learners and teachers - by introducing a new partner. There is also a host of secondary impacts that are social, involving the larger contexts in which teaching and learning take place.

Our knowledge base regarding how to exploit this inevitable force in education is very thin, as we are in the first decades of its appearance. In challenging most traditional assumptions about teaching and learning, technology forces us to think deeply about all aspects of our work, including the forms of the research that need to be undertaken to use it to best advantage. Clearly, our most important work lies ahead of us.

AKNOWLEDGEMENTS

We would like to thank Celia Hoyles and Liddy Nevile who acted as reviewers for this chapter, as well as all those who responded to our requests for information and advices: Rolf Biehler, Daniel Chazan, Tommy Dreyfus, Joel Hillel, Jean-Marie Laborde, Rosamund Sutherland.

REFERENCES

Abelson, H. & diSessa, A.: 1981, *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*, MIT Press, Cambridge.

Allen, R., Desmoulins, C. & Trilling, L.: 1992, 'Tuteurs intelligents et Intelligence Artificielle: problèmes posés en construction de figures géométriques', in C. Frasson (ed.), *Intelligent Tutoring Systems*, Springer Verlag, Berlin, 325-334.

Anderson, J. R., Boyle, C. F., Corbett, A. T. & Lewis M. W.: 1990, 'Cognitive Modeling and Intelligent Tutoring', *Artificial Intelligence* 42 (1), 7-49.

Anderson, J.R., Boyle, C.E., & Yost G.: 1985, 'The Geometry Tutor', *Prodeedings of the International Joint Conference on Artificial Intelligence*, Los Angeles, 1-7.

[4] Recently have appeared on the market calculators offering all the type of mathematical environments that we have mentioned here (CAS, cabri-like software, spreadsheet, etc.), even sometimes with specific pedagogical extensions.

Arsac, G., Balacheff, N., & Mante, M.: 1992, 'Teacher's Role and Reproducibility of Didactical Situations', *Educational Studies in Mathematics* 23 (1), 5-29.

Artigue, M.: 1996, 'Le logiciel "DERIVE" comme révélateur de phénomènes didactiques liés à l'utilisation d'environnements informatiques pour l'apprentissage', *Educational Studies in Mathematics*, to appear.

Artigue, M.: 1992, 'Didactic Engineering', *Recherches en didactique des mathématiques* (Selected Papers), 41-66.

Bachelard, G.: 1984, *The New Scientific Spirit.* Trans. A. Goldhammer, Beacon Press, Boston.

Balacheff, N.: 1991, 'Treatment of Refutations : Aspects of the Complexity of a Constructivist Approach of Mathematics Learning', in E. Von Glasersfeld (ed.), *Radical Constructivism in Mathematics Education*, Kluwer Press Publisher, Dordrecht, 89-110.

Balacheff, N.: 1993, Artificial Intelligence and Real Teaching', in C. Keitel and K. Ruthven (eds.) *Learning Through Computers: Mathematics and Educational Technology*, Springer Verlag, Berlin, 131-158.

Balacheff, N.: 1996, 'Advanced Educational Technology: Knowledge Revisited', in T. Liao (ed.), *Advanced Educational Technology: Research Issues and Future Potential*., Springer Verlag, Berlin, in press.

Balacheff, N. & Sutherland R.: 1994, 'Epistemological Domain of Validity of Microworlds. The Case of Logo and Cabri-géomètre', in R. Lewis and P. Mendelsohn (eds.), *Lessons from Learning*, North-Holland/Elsevier Science B. V., Amsterdam , IFIP Transactions A46, 137-150.

Bellemain, F. & Capponi, B.: 1992, 'Spécificité de l'organisation d'une séquence d'enseignement lors de l'utilisation de l'ordinateur', *Educational Studies in Mathematics* 23(1), 59-97.

Biehler, R.: 1993: 'Software Tools and Mathematics Education: The Case of Statistics', in C. Keitel and K. Ruthven (eds.), *Learning Through Computers: Mathematics and Educational Tedchnology*, Springer Verlag, Berlin, 68-100.

Boero, P.: 1992, 'The Crucial Role of Semantic Field in the Development of Problem Solving Skills in the School Environment', in J. P. Ponte, J. F. Matos, J. M. Matos and D. Fernandes (eds.), *Mathematical Problem Solving and New Information Technologies*, Springer Verlag, Berlin, 77-91.

Brousseau, G.: 1986, 'Fondements et méthodes de la didactique des mathématiques', *Recherches en didactique des mathématiques* 7(2), 33-115.

Brousseau, G.: 1992, 'Didactique: What it Can Do for the Teacher', *Recherches en didactique des mathématiques* (Selected Papers), 7-39.

Brown, J. S.: 1983, 'Processus Versus Product: A Perspective on Tools for Communal and Informal Electronic Learning' Reprinted in *Journal of Educational Computing Research*, 1985, 1, 179-201.

Brown, J. S. & Burton, R. R.: 1978, 'Diagnostic Models for Procedural Bugs in Basic Mathematical Skills', *Cognitive Science* 2, 155-191.

Burton, R. R. & Brown, J. S.: 1979, 'An Investigation of Computer Coaching for Informal Learning Activities', *International Journal of Man-Machine Studies* 11, 5-24.

Capponi, B.: 1993, 'Modification des menus dans Cabri-géomètre : des symétries comme outil de conctruction', *Petit X* 33, 37-68.

Casti, J.: 1992, *Reality Rules* (two volumes), Wiley Interscience, New York.

Chan T. W.: 1993, ‘Some Techniques for Building Mathematical Intelligent Tutoring Systems’, in H. S. Nwana (ed.), *Mathematical Intelligent Learning Environments*, Intellect, Oxford, UK, 52-78.

Chazan, D.: 1990, ‘Quasi-empirical Views of Mathematics and Mathematics Teaching’, *Interchange* 21 (1), 14-23.

Chazan, D. & Houde, R.: 1989, *How to Use Conjecturing and Microcomputers to Teach Geometry*, National Council of Teachers of Mathematics, Reston, VA.

Chazan, D. & Yerushalmy, M.: 1995, ‘Charting a Course fo Secondary Geometry’, in R. Lehrer and D. Chazan (eds.), *New Directions in the Teaching and Learning of Geometry*, Erlbaum, Hillsdale, pre-print.

Chevallard, Y.: 1985, *La transposition didactique* (nouvelle édition revue et augmentée, 1991), Editions La Pensée Sauvage, Grenoble.

Confrey, J.: 1990, ‘A Review of the Research on Student Conceptions in Mathematics, Science and Programming’, in Cazden and Courtney (eds.), *Review of Research in Mathematics Education. American Educational Research Association* 16, 3-56.

Cornu, B. (ed.): 1992, *L’odinateur pour enseigner les mathématiques*, Presses Universitaires de France, Paris.

Derive (software): 1994, Software Warehouse, Honolulu, HI.

Devaney, R.: 1992, *A first course in chaotic dynamical systems*, Addison-Wesley, Reading, MA.

Dreyfus, T.: 1993, ‘Didactic Design and Computer-based Learning Environments’, in C. Keitel and K. Ruthven (eds.), *Learning Through Computers: Mathematics and Educational Tedchnology*, Springer Verlag, Berlin, 101-130.

Du Boulay, B.: 1978, *Learning primary mathematics through computer programming*, Ph. D. Thesis, University of Edinburgh.

Edwards, L. D.: 1989, *Children's Learning in a Computer Microworld for Transformation Geometry*, Doctoral Dissertation, University of California at Berkeley, Berkeley.

Elsom-Cook, M.: 1990, *Guided Discovery Tutoring*, Paul Chapman Publishing, London.

Fey, J. T.: 1984, *Computing and mathematics: The impact on secondary school curricula*, National Council of Teachers of Mathematics, Reston, VA.

Fey, J. T.: 1989a, ‘Technology and Mathematics Education: A Survey of Recent Developments and Important Problems’, *Educational Studies in Mathematics* 20, 237-272.

Fey, J. T.: 1989b, ‘School Algebra for the Year 200’, in S. Wagner and C. Kieran (eds.), Research Issues in the Learning and Teaching of Algebra, National Council of Teachers of Mathematics, Reston, VA, 199-219.

Gleick, J.: 1987, *Chaos: Making a new science*, Viking Penguin, New York.

Goldenberg, E. P. & Cuoco, A.: 1996, ‘What is Dynamic Geometry?’, in R. Lehrer and D. Chazan (eds.), *New Directions in the Teaching and Learning of Geometry*, Erlbaum, Hillsdale, pre-print.

Guin, D.: 1991, ‘Modélisation de la démonstration géométrique dans Geometry Tutor’, *Annales de Didactique et de Sciences Cognitives* 4, IREM de Strasbourg, Strasbourg, 5-40.

Hancock, C.: 1995, *Tabletop* (software), Broderbund Inc., Santa Barbara, CA.

Hancock, C., Kaput, J. & Goldsmith, L.: 1992, ‘Authentic Inquiry with Data: Critical Barriers to Classroom Implementation’, *Educational Psychologist* 27(3), 337-364.

Hillel, J.: 1993, 'Computer Algebra Systems as Cognitive Technologies: Implication for the Practice of Mathematics Education', in C. Keitel and K. Ruthven (eds.), *Learning Through Computers: Mathematics and Educational Tedchnology*, Springer Verlag, Berlin, 18-47.

Hillel, J. & Kieran, K.: 1987, 'Schema Used by 12-year-olds in Solving Selected Turtle Geometry Tasks', *Recherches en didactique des mathématiques* 8(1/2), 61-102.

Hofstadter, D. R.: 1981, 'Strange Attractors: Mathematical Patterns Delicately Poised Between Order and Chaos', *Science* 245, 5-22.

Hoyles, C.: 1993, 'Microworlds/Schoolworlds: The Transformation of an Innovation', in C. Keitel and K. Ruthven (eds.), *Learning Through Computers: Mathematics and Educational Tedchnology*, Springer Verlag, Berlin, 1-17.

Hoyles, C.: 1995, 'Thematic Chapter: Exploratory Software, Exploratory Cultures?', in A. A. diSessa , C. Hoyles and R. Noss (eds.), *Computers and Exploratory Learning,* Springer Verlag, Berlin, 199-219.

Hoyles, C. & Noss, R.: 1992, 'A pedagogy for Mathematical Microworlds', *Educational Studies in Mathematics* 23(1), 31-57.

Hoyles, C. & Sutherland, R.: 1989, *Logo Mathematics in the Classroom.*, Routledge, London.

Kang, W. & Kilpatrick, J.: 1992, 'Didactics Transposition in Mathematics TextBooks', *For the Learning of Mathematics*, 12(1), 2-7.

Kaput, J.: 1993, *MathCars* (Computer animation video available from the author), University of Massachusetts, Dartmouth.

Kaput, J.: 1992, 'Technology and mathematics education', in D. Grouws (ed.), *Handbook on research in mathematics teaching and learning*, Macmillan, New York, NY, 515-556.

Kaput, J.: 1994, 'Democratizing Access to Calculus: New Routes Using Old Roots', in A. Schoenfeld (ed.) *Mathematicical Thinking and Problem Solving*, Erlbaum, Hillsdale, NJ, 77-156.

Kaput, J.: 1995, *SimCalc: Simulations for Calculus Learning*, Year 2 Annual Report to the National Science Foundation (available from the author), University of Massachusetts, Dartmouth.

Kaput, J. & Nemirovsky, R.: 1995, 'Moving to the Next Level: A Mathematics of Change Theme Throughout the K-16 Curriculum', *UME Trends* 6(6), 20-21.

Kaput, J. & Thompson, P.: 1994, 'Technology in Mathematics Education: The First 25 Years in the JRME', *Journal for Research in Mathematics Education* 25(6), 676-684.

Kaput, J., Upchurch, R. & Burke, M.: 1996, *Cybernetic Manipulatives for Additive Structures* (computer program), University of Massachusetts at Dartmouth, Dartmouth, MA.

Kauffman, S.: 1993, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York.

Kaye, A. R.: 1992, *Collaborative Learning Through Computer Conferencing*, Springer Verlag, Berlin.

Kilpatrick, J. & Davis, R. B.: 1993, 'Computers and Curriculum Change in Mathematics', in C. Keitel and K. Ruthven (eds.), *Learning Through Computers: Mathematics and Educational Tedchnology*, Springer Verlag, Berlin, 203-221.

Kimball, R.: 1973, *Self-optimizing Computer-Assisted Tutoring: Theory and Practice*, Psychology and Education series, Technical report 206, Stanford University, Stanford CA.

Klotz, G. & Jakiw, N.: 1988, *The Geometer's Sketchpad* (software), Key Curiculum Press, Berkeley, CA.

Laborde, C.: 1992a, 'Enseigner la géométrie : permanences et révolutions', in C. Gaulin, B. R. Hodgson, D. H. Wheeler and J. C. Egsgard (eds.) *Proceedings of the 7th International Congress on Mathematical Education*, Les Presses de l'Université de Laval,. Québec, 47-75.

Laborde, C.: 1992b, 'Solving Problems in Computer Based Geometry Environments: The Influence of the Features of the Software', *Zentrablatt für Didactik des Mathematik* 92 (4), 128-135.

Laborde, C.: 1993, 'The Computer as Part of the Learning Environment: the Case of Geometry', in C. Keitel and K. Ruthven (eds.), *Learning Through Computers: Mathematics and Educational Tedchnology*, Springer Verlag, Berlin, 48-67.

Laborde, C. & Capponi, B.: 1994, 'Cabri-géomètre, constituant d'un milieu pour l'apprentissage de la notion de figure géométrique', in N. Balacheff and M. Vivet (eds.), *Didactique et Intelligence Artificielle*, Editions La Pensée Sauvage, Grenoble, 165-210.

Laborde, J.-M.: 1994, 'Intelligent Microworlds and Learning Environments', in J.-M. Laborde (ed.), *Intelligent Learning Environments: The case of Geometry,* Springer Verlag, Berlin, in press.

Laborde, J.-M.: 1996, 'On Models for Exploring Non Euclidean Geometry in a Dynamic Geometry Environment Like Cabri-géomètre', in J. King (ed.), *Dynamic Geometry*, MAA Edition, to appear.

Laborde, J.-M.: 1985, 'Projet de géométrie', in D. Guin, F.-F. Nicaud and D. Py (eds.) Environnements Interactifs d'Apprentissage avec Ordinateur, Eyrolles, Paris, 1995, 41.

Maple (software): 1990, Brooks/Cole Publishing Co, Pacific Grove, CA.

Math Blaster (software): 1993, Davidson & Associates, Los Angeles, CA.

McCalla, G.: 1994, *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, Springer-Verlag, Berlin.

Moler, C.: 1995, *MatLab* (software), Mathworks Inc., Wellesley, MA.

Nemirovsky, R.: 1992, *Students Making Sense of Chaotic Behavior* (manuscript available from the author), Technical Education Resource Center, Cambridge, MA.

Nicaud, J.-F.: 1992, 'A General Model of Algebraic Problem Solving for the Design of Interactive Learning Environments', in J. P. Ponte, J. F. Matos, J. M. Matos and D. Fernandes (eds.), *Mathematical Problem Solving and New Information Technologies*, Springer Verlag, Berlin, 267-285.

Ohlsson, S.: 1987, 'Sense and Reference in the Design of Interactive Illustrations for Rational Numbers', in R. Lawler and M. Yazdani (eds.), *Artificial Intelligence and Education: Learning Environments and Tutoring Systems* (Vol. 1), Ablex Publishing Corporation, Norwood, NJ, 307-344.

Papert, S.: 1973, 'Use of Technology to enhance education', *AI Memo* (n°298), MIT, Cambridge, MA.

Papert, S.: 1980, *Mindstorms*, Basic Books, New York, NY.

Parzysz, B.: 1988, 'Knowing Versus Seing: Problems of the Plane Representation of Space Geometry Figures', *Educational Studies in Mathematics* 19(1), 79-92

Payan, C.: 1992, *Cabri-géomètre: du continu au discret*., Document interne, IMAG, Grenoble.

Py, D. & Nicolas, P.: 1990, 'Mentoniezh: a Geometry I.T.S. for Figure Drawing and Proof Setting', *Journal of Artificial Intelligence and Education* 1(3), 41-56.

Resnick, M.: 1994, *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*, MIT Press, Cambridge, MA.

Roschelle, J.: 1994, 'Designing for cognitive communication: The case of mental models in qualitative physics', *Electronic Journal of Virtual Culture*, 2(2) [anonymous ftp byrd.mu.wvnet.edu]

Rubin, A. & Bruce, B.: 1991, *Statistics Workshop* (software), Wings for Learning/Sunburst, Scotts Valley, CA.

Sandefur, J.: 1990, *Discrete Dynamical Systems: Theory and Applications*, Clarendon Press, Oxford.

Sandefur, J.: 1992, 'Technology, Linear Equations, and Buying a Car' *The Mathematics Teacher* 85 (7), 562-567.

Sandefur, J.: 1993, *Discrete dynamical modeling*, Oxford University Press, Oxford.

Schwartz, J. & Yerushalmy, M.: 1984, *The Geometric Supposer: Triangles*, Sunburst Communication, Pleasantville, NY.

Sleeman, D. H.: 1982, 'Inferring (mal) rules from pupils' protocols', *Proceedings of the European Conference on Artificial Intelligence*, Orsay, France, 160-164.

Steen, L. (ed.): 1990, *On the shoulders of gaints: New approaches to numeracy*, National Academy Press, Washington, DC.

Steffe, L. P. & Wiegel, H. G.: 1994, 'Cognitive Play and Mathematical Learning in Computer Microworlds', *Educational Studies in Mathematics*, 26(2/3), 111-134.

Sutherland, R.: 1993, 'Connecting Theory and Practice: Results from the Teaching of Logo', *Educational Studies in Mathematics* 24, 1-19.

Sutherland, R. & Balacheff, N.: 1996, *Didactical Complexity of Computational Environments for the Learning of Mathematics*, Unpublished manuscript, IMAG, Grenoble.

Sutherland, R. & Rojano, T.: 1993, A Spreadsheet Approach to Solving Algebra Word Problems, *Journal of Mathematical Behaviour* 12(4), 351-383.

Tahri, S.: 1993, *Modélisation de l'interaction didactique : un tuteur hybride sur Cabri-géomètre pour l'analyse de décisions didactiques*, Thèse, Université Joseph Fourier, Grenoble.

Thompson, P. W.: 1987, 'Mathematical Microworld and Intelligent Computer Assisted Instruction', in G. E. Kearsley (ed.), *Artificial Intelligence and Instruction: Applications and Methods*, Addison-Wesley, New York, 83-109.

Thompson, P. W.: 1992, 'Notations, Conventions, and Constraints: Contributions to Effective Uses of Concrete Materials in Elementary Mathematics, *Journal for Research in Mathematics Education* 23(2), 123-147.

Thompson, P. W.: 1993, 'Quantitative Reasoning, Complexity, and Additive Structures', *Educational Studies in Mathematics* 25(3), 165-208.

Trgalova, J.: 1995, *Etude historique et épistémologique des coniques et leur implémentation informatique dans le logiciel Cabri-géomètre*, Thèse, Université Joseph Fourier, Grenoble.

Vivet, M.: 1992, 'Uses of ITS: Which role of the Teacher?', in E. Costa (ed.) *New Direction for Intelligent Tutoring Systems*, Springer Verlag, Berlin, 171-180.

Wenger, E.: 1987, *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann Publishers Inc., Los Altos, CA.

White, B. Y. & Fredericksen, J. R.: 1986, *Progressions of Qualitative Models as a Foundation for Intelligent Learning Environments* (Research Report n°6277), BBN Laboratories, Cambridge, MA.

White, B.: 1993, Thinkertools: Casual models, conceptual change, and science education. *Cognition and Instruction* 10, 1-100.

Wolfram, S.: 1995, *Mathematica*, Wolfram Associates, Urbana, IL.

Yerushalmy, M., & Schwartz, J.: 1993, 'Seizing the opportunity to make algebra mathematically and pedagogically interesting', in T. Romberg, E. Fennema and T. Carpenter (eds.), *Integrating Research on the Graphical Representation of Functions*, Lawrence Erlbaum Associates, Hillsdale, NJ, 41-68.