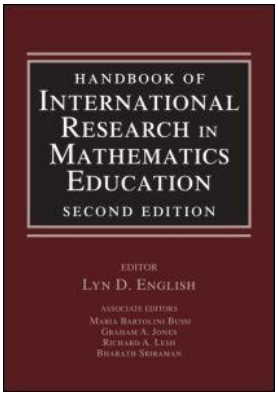

Handbook of International Research in Mathematics Education

Lyn D. English, Maria Bartolini Bussi, Graham A. Jones, Richard A. Lesh, Bharath Sriraman, Dina Tirosh

Developing new notations for a learnable mathematics in the computational era

Publication details
https://www.routledgehandbooks.com/doi/10.4324/9780203930236.ch26
James Kaput, Richard Noss, Celia Hoyles
Published online on: 19 Jun 2008

How to cite :- James Kaput, Richard Noss, Celia Hoyles. 19 Jun 2008, *Developing new notations for a learnable mathematics in the computational era from:* Handbook of International Research in Mathematics Education Routledge
Accessed on: 02 Dec 2023
https://www.routledgehandbooks.com/doi/10.4324/9780203930236.ch26

Section 4

Influences of advanced technologies

This page intentionally left blank

26 Developing new notations for a learnable mathematics in the computational era

James Kaput
University of Massachusetts-Dartmouth

Richard Noss and Celia Hoyles
Institute of Education, University of London

Since the publication of this chapter, Jim Kaput was killed in a tragic road accident. We have all lost an energetic, visionary, and dedicated colleague. Mathematics education research has lost one of its greatest exponents, a researcher who not only understood the world of education, but knew how to change it. His theoretical work on the evolution of notational systems—much of it presented in this chapter—as well as his practical contribution, most recently through SimCalc, is living testimony to the importance and impact of his work. And we have lost a friend: a friend with a wonderful sense of humor and wit, a vibrant sense of fun, and an inspired intelligence. We will miss him.

Not for the first time, we are at a turning point in intellectual history. The appearances of new computational forms and literacies are pervading the social and economic lives of individuals and nations alike. Yet nowhere is this upheaval correspondingly represented in educational systems, in classrooms, or in school curricula. As far as mathematics is concerned, the massive changes to mathematics that characterize the late 20th century—in terms of the way it is done, and what counts as mathematics—are almost invisible in the classrooms of our schools and, to only a slightly lesser extent, our universities.

The real changes are not technical: they are cultural. Understanding them (and why some things change quickly and others change slowly) is a question of the social relations among people, not among things. Nevertheless, there are important ways in which computational technologies are different from those that preceded them, and in trying to assess the actual and potential contribution of these technologies to education, it will help to view them in a historical light.

The notation systems we use to present and re-present our thoughts to ourselves and to others, to create and communicate records across space and time, and to support reasoning and computation, constitute a central part of any civilization's infrastructure. As with infrastructure in general, it functions best when it is taken for granted, invisible, when it simply works. The original chapter was prepared as the UK ground transportation system had, due to a number of additive causes, almost totally failed, and when the electricity production and distribution system of California was likewise in disarray. When the infrastructure either fails or undergoes changes, the disruptions can be major. Further, changes tend to propagate, so that one change causes another in tightly interconnected systems—when the electricity goes out, lots of other things go out too. The same is true on the positive side—when a new technological infrastructure appears, such as the Internet, many things change, often in unpredictable ways as sequences of new opportunity spaces open up and old ones close down. Entire industries are born, old ways of doing things change, sometimes in fundamental ways—how people participate in the economy changes, the physically-based means for defining and controlling

ownership of intellectual property are challenged, and indeed, the means by which innovation itself is fostered changes. Lastly, these kinds of infrastructural changes are typically not the result of systematic planning or central control. They emerge in unpredictable ways from the mix of existing circumstances.

These general questions of representational infrastructures may seem far removed from the apparently more mundane task of learning mathematics; but the central challenge of mathematical learning for educators is surely the design of learnable systems. Such systems depend for their learnability (or lack of it) on the particularities and interconnectedness of the representational systems in which they are expressed. These, as we stated at the outset, are undergoing rapid change. In order to understand these changes more fully, we wish to examine the longer-term sweep of representational infrastructure change across several important examples in order to provide a long-term perspective on the content choices and trends embodied in school mathematics. And as we shall see, mathematics enjoys a particularly interesting role in this story.

THE EARLIEST QUANTITATIVE NOTATION SYSTEMS

Most representational infrastructures develop in response to the social needs of one or more groups, where the needs might be very broad and involve the whole society, as was the case with the development of writing, or they may serve a smaller subgroup such as mathematicians and scientists, who needed to express and reason with general quantitative relationships and hence developed what we now know as the algebraic system. Indeed, the earliest, prephonetic written language co-evolved with mathematics in the cradle of western civilization some 6–8 thousand years ago to record physical quantities through a gradual process of semiotically abstracting the physical referents into systems of schematic representations of those referents (Kaput & Shaffer, 2002).

The systematicity initially took the form of separating inscriptions denoting object-types from inscriptions denoting their properties (identity of owner, size, color, vintage, etc.), and, gradually over hundreds of years, the numerosity property. Inscriptions denoting numerosity gradually condensed, from a repetition model where four instances of an item were represented by four tokens for that type of item, then four tallies adjacent to a single token for that type of item, to a modifier model employing symbols denoting numerosity, that is, a symbol for "four," replacing the tally marks. This last step required the co-evolution of the concept of counting number, mainly through the work of those specialists who were the scribes responsible for producing the records. There is little indication that such early numbering systems were used for computation other than incrementing and decrementing.

Distinct from, but certainly not unrelated to the notational system, was the physical system in which it was instantiated: primarily clay, which afforded the means to impress tokens of objects (sheepskins, jars of olive oil, etc.) first onto clay envelopes containing the tokens and then tablets, when the tokens came to be regarded as redundant (Schmandt-Besserat 1978, 1992). The medium was temporarily inscriptable, and then hardened to provide stability that enabled the inscriptions to act as records, indeed, somewhat mobile records. In this way, evolutionary limitations of human biological memory were finally overcome through the use of "extracortical" records (Donald, 1991).

THE EVOLUTION OF NOTATION SYSTEMS SUPPORTING QUANTITATIVE COMPUTATION

We now examine the evolution of a second representational infrastructure. In the several millennia that followed, and across several different societies where urbanization and commerce

developed, various number systems evolved to support ever better and more compact ways of expressing quantities and abstract numbers, particularly to express the large numbers required for calendar purposes and for tracking quantities in the city-states and empires—Babylonian, Egyptian, and eventually Roman. Importantly for our purposes, while they typically embodied grouping structures, they tended not to be rigorously positional and tended not to be fully hierarchical. The most tightly structured and efficient system was the Babylonian (semi) sexigesimal (base 60) system. It employed a mix of additive and multiplicative methods of representing numbers as there were no common symbols for smaller numbers (as with the Hindu-Arabic numerals). It did, however, use position to denote powers of 60. Hence a number would be represented as an array of symbols for units, 10, and powers of 60 (using cuneiform, or wedge-based signs). As is widely appreciated, this system supported a rich practical mathematics that served many aspects of society for more than two millennia, although with the lack of a zero for a placeholder (in its later years a placeholder system did develop), it did not support efficient multiplication or division. Further, the lack of compact numerals for the first nine numbers meant that it was considerably less efficient for writing numbers in the hundreds and thousands than our current system, and even less efficient, relatively, for larger numbers. Of course, the contemporaneous writing systems were likewise ideographic and difficult to learn and hence the tool of specialists—scribes (Walker, 1987). The later Roman system was less structured and less multiplicative in its organization, and hence even less efficient for multiplication and division.

How did these systems survive in supporting the quite extensive calculational tasks they were called upon to serve? The answer is clear: only a very small minority of the respective members of these societies were needed to do such calculations, and these scribes were specially trained in the art of manipulating the symbol systems. In this respect, the role of the physical medium (e.g., the marks made on clay and so on) were crucial in supporting the prodigious amounts of human processing power that would otherwise be engaged. While the structural features of the notational system were not particularly tuned to calculational ends (anyone who has ever tried to do long division with Roman numerals will testify to this) the combination of the physical instantiation of symbols, *together with human processing power on the part of a few,* was sufficient to sustain powerful empires over hundreds and thousands of years. Furthermore, the existing static record-keeping capacity could be used to record methods, results (especially in the case of the Babylonians, who made wide use of tables of all sorts to record quantitative information and mathematical relationships, make astronomical predictions, and so on), and even instructional materials by which expertise could be extended across generations (Kline 1953).

Another big representational transformation had roots several centuries earlier, in the 8th to 11th centuries, preceding Fibonacci's importation of the Hindu-Arabic numerals into Europe in the early 13th century. More efficient methods of computing developed, based on systematic use of specially-marked physical "counting tables" on which physical tokens were manipulated. In this way, the technology of the counting tables externalized some of the knowledge and transformational skill which would otherwise have existed only in the minds of individuals: the physical instantiation of these skills directly supported, not only the limited processing capacity of human brains, but the affordance of the notational system for achieving results.

These results of computations were recorded first in Roman numerals, but then gradually more often in Hindu-Arabic numerals. These methods are typically referred to as "abacus" style computations based on the Greek word for slab, on which the procedures took place. At the same time, and then more intensely during the 13th century, new and more efficient ways of computing evolved based on manipulation of the readily inscriptable Hindu-Arabic numerals in the positional and hierarchical number system that Fibonacci had described. These were referred to as "algorithm" style computations based on a Latinized version of the name Abu Ja'far Muhammed ibn Musa al-Khwarizmi, a mathematician from Baghdad who wrote

an arithmetic book describing some of the early computational schemes using Hindu-Arabic numerals.

It is instructive to see the prominence of commercial needs of the day in his description of algebra:

> ... what is easiest and most useful in arithmetic, such as men constantly require in cases of inheritance, legacies, partition, lawsuits, and trade, and in all their dealings with one another, or where the measuring of lands, the digging of canals, geometrical computations, and other objects of various sorts and kinds are concerned. F. Rosen (Trans.), *Muhammad ibn Musa Al-Khwarizmi: Algebra* (London, 1831)

Clearly, al-Khwarizmi conceived algebra as a way of solving pressing practical problems of the Islamic Empire. Similarly, in response to burgeoning commerce in the 14th and 15th centuries in northern Italy and elsewhere, the algorithms were refined and gradually displaced the abacus methods, although not without controversy.[1] The efficiency payoff of a positional and exponentially hierarchical system was enormous because it allowed a person to compute simply by writing and rewriting the small set of ten symbols according to certain rules (the algorithms) and, on the basis of the quantitative coherence of the notation system, be assured of a correct answer based on the rules alone. *Computational skill became encoded in syntactically defined rules on a symbol system.*

The algorithmic methods were put forward (anonymously) in what amounts to the first arithmetic text, the *Treviso*, named after the city outside Venice where it was published in 1478, less than 40 years after Gutenberg's introduction of moveable type (itself a response to the pressing need to find a way of salvaging religious texts which contained mistakes, without destroying the entire work). These algorithms, exploiting the physical positional structure of the notation system and the paper medium, are essentially the same forms for addition, subtraction, multiplication, and division that have dominated school mathematics to the 20th century. In the book (translated and appearing in Swetz, 1987), they were illustrated within the contexts of commerce and currency exchange, and were passed along from generation to generation as a body of practical knowledge in what amounted to professional schools for "reckoners"—the accountants of the time.

Interestingly, the *Treviso* was written in the vernacular, as opposed to Latin, and thus was one of the first printed mathematics books intended to serve a "non-academic" public—or at least that public who needed to know these special techniques. The new representational infrastructure helped democratize access to what had previously been the province of a small, intellectual elite, since up to that time numerical computations beyond addition and simple subtraction were a scholarly pursuit undertaken at the universities. Recall the oft-cited anecdote from Dantzig (1954):

> It appears that a [German] merchant had a son whom he desired to give an advanced commercial education. He appealed to a prominent professor of a university for advice as to where he should send his son. The reply was that if the mathematical curriculum of the young man was to be confined to adding and subtracting, he perhaps could obtain the instruction in a German university; but the art of multiplying and dividing, he continued, had been greatly developed in Italy, which, in his opinion, was the only country where such advanced instruction could be obtained. (p. 26)

Indeed, the question of inclusion of these same algorithms in school mathematics, to support basic shopkeeper arithmetic, currency exchange, and other simple arithmetic tasks, continued to be the subject of vigorous debate through the 20th century and even into the 21st. Of course, in the intervening 5 centuries, the practical role of arithmetic has broadened with the increasing complexity of modern life, especially in the workplace, and it has assumed a cor-

nerstone position in school mathematics. Indeed, the skills of arithmetic are seen, in almost all developed societies, not only as essential for the efficient operation of economies, but as an entitlement of an educated individual. We shall return to this issue below, but for the moment, it is worth delineating two separate skills that arithmetic teaching in the twentieth century came to serve: the one concerned with obtaining answers quickly and correctly, and the other as a backdrop against which the process of executing algorithms could be performed, number relationships learned, and manipulative methods practiced. While execution was the preserve of the human mind, this distinction hardly arose, but as we shall see it becomes more central in this computational era.

THE EVOLUTION OF ALGEBRAIC NOTATIONS

We now examine a third example of a representational infrastructure. Algebra began in the times of the Egyptians in the second millennium BC, as evidenced in the famous Ahmes Papyrus, by using available writing systems to express quantitative relationships, especially to "solve equations"—to determine unknown quantities based on given quantitative relationships. This is the so-called "rhetorical algebra" that continued to Diophantus' time in the 4th century of the Christian era, when the process of abbreviation of natural language statements and the introduction of special symbols began to accelerate. Algebra written in this way is normally referred to as "syncopated algebra." By today's standards, achievement to that point was primitive, with little generalization of methods across cases and little theory to support generalization. Indeed, approximately 2 millennia produced solutions to what we would now refer to as linear, quadratic, and certain cubic equations (of course, they were not written as equations), often based in contrived and stylized concrete situations, and not much more. Indeed, it appears that, in the absence of a systematic symbol system, the stylized situation provided a kind of semi-abstract conceptual scaffolding for the quantitative reasoning that constituted the methods. The accumulated skill was encoded in illustrative examples rather than in syntactically defined rules for actions on a symbol system.

Then, in a slow, millennium-long struggle involving the co-evolution of underlying concepts of number (see especially Klein, 1968), algebraic symbolism gradually freed itself from written language in order to support techniques that increasingly depended on working with the symbols themselves according to systematic rules of substitution and transformation—rather than the quantitative relations for which they stood. Just as the symbolism for numbers evolved to yield support for rule-based operations on inscriptions taken to denote numbers, so the symbolism for quantitative *relations* likewise developed. Bruner (1973) refers to this as an "opaque" use of the symbols rather than "transparent" use: the former implies attention to actions on the inscriptions, while the latter implies that actions are guided by reasoning about the entities to which the inscriptions are assumed to refer.

In effect, algebraic symbolism gradually freed itself from the (highly functional) ambiguities and general expressiveness of natural language. The newly developed and systematic semiotic structures embodied hard-won understandings of general mathematical relations and, by the 17th century, functions. This symbol system also embodied forms of generality, (particularly through the use of symbols for variables), and the dual use of operation symbols (so symbols such as "+" could be applied to symbols for variables ranging over sets of numbers as well as for the numbers themselves). Hence general statements of quantitative relations could be expressed efficiently.

However, the more important aspects of the new representational infrastructure are those that involved the rules, the syntax, for guiding operations *on* these expressions of generality. These emerged in the 17th century as the symbolism became more compact and standardized in the intense attempts to mathematize the natural world that reached such triumphant fruition in the "calculus" of Newton and Leibniz. In the words of Bochner (1966):

> Not only was this algebra a characteristic of the century, but a certain feature of it, namely the "symbolization" inherent to it, became a profoundly distinguishing mark of all mathematics to follow.... (T)his feature of algebra has become an attribute of the essence of mathematics, of its foundations, and of the nature of its abstractness on the uppermost level of the "ideation" à la Plato. (pp. 38–39)

Beyond this first aspect of algebra, its role in the expression of abstraction and generalization, Bochner also pointed out the critical new ingredient:

> ... that various types of 'equalities,' 'equivalences,' 'congruences,' 'homeomorphisms,' etc. between objects of mathematics must be discerned, and strictly adhered to. However this is not enough. In mathematics there is the second requirement that one must know how to 'operate' with mathematical objects, that is, to produce new objects out of given ones. (p. 313)

Indeed, Mahoney (1980) points out that *this development made possible an entirely new mode of thought* "characterized by the use of an operant symbolism, that is, a symbolism that not only abbreviates words but represents the workings of the combinatory operations, or, in other words, a symbolism with which one operates" (p. 142).

This second aspect of algebra, the syntactically guided transformation of symbols while holding in abeyance their potential interpretation, flowered in the 18th century, particularly in the hands of such masters as Euler. At the same time, this referral of interpretation led to the further separation of algebraic and natural language writing and hence the separation from the phonetic aspects of writing that connect with the many powerful narrative and acoustic memory features of natural language. Indeed, as is well known from such examples as the "Students-Professors Problem" (Clement, 1982; Kaput & Sims-Knight, 1983), the algebraic system can be in partial *conflict* with features of natural language.

Thus, over an extremely long period, a new special-purpose operational representational infrastructure was developed that reached beyond the operational infrastructure for arithmetic. However, in contrast with the arithmetic system, *it was built by and for a small and specialized intellectual elite* in whose hands, quite literally, it extended the power of human understanding far beyond what was imaginable without it. In the hands of an extremely small community over the next 250 years, the expressive and operational aspects of this narrowly-scoped representational infrastructure made possible a science and technology that irreversibly changed the world, as well as of our views of it and of our place in it.

CALCULUS AND THE IDEA OF "A CALCULUS"

Our last example of representational infrastructure evolution involves calculus. The notion of an automatic computing machine to carry out numeric calculations, as we have seen, is very old. Leibniz, however, wanted to go further and be able to compute logical consequences of assumptions through an appropriate symbol system. He understood, perhaps more clearly than anyone before him, not only that choice of notation system was critically important to what one could achieve with the system, but also and more specifically, that a well-chosen syntax for operations on the notation system could support ease of symbolic computation. Hence, as reflected in correspondence with contemporaries, especially Huygens (Edwards, 1979), he was very careful in the design of a notation to represent his findings regarding how a function was related to what we now call its derivative or integral. His goal was that his new notation would support a "calculus" for computing such new functions in the general sense that the word "calculus" was used in those times.[2] His nicely compact and mnemonic

notation also allowed a direct expression of the relations between derivatives and integrals, relations expressed in the Fundamental Theorem of Calculus.

Indeed, diSessa (2000), reminds us that Leibniz' notation, which dominated the way calculus was used more than 300 years later, was at least as important as the insights that it encoded. After all, these ideas were also created by Newton. But Newton's brilliant insights and methods have come to be learned and used for generations in Leibniz' notation, and the reluctance of his British followers to adopt Leibniz' notation was likely a significant factor in the century-long lag of British mathematics behind that of the Continent (Boyer, 1959; Edwards, 1979). diSessa points out that Leibniz' notation became "infrastructural" (p. 11)—in the same sense that we have been using the phrase "representational infrastructure" in this chapter. Incidentally, it is pointed out by diSessa that the achievement of infrastructural status for Leibniz's notation was in no small part due to the fact that it was easier to teach.

So once again, as in the case of arithmetic and then algebra, the development of a compact, efficient notation system turned out to be a critical factor in what followed.

COMPUTATIONAL MEDIA AND THE SEPARATION OF OUTCOME FROM PROCESS

The foregoing provides a brief overview of the structural changes in the semiotics of mathematical expression over time, leading to the emergence of complex and strongly supportive systems which sustain and expand the possibilities of human calculation and manipulation—at least for the few who were inducted into its use. We will argue that there are two key developments in a computational era: first that human participation is no longer required for the *execution of a process* and second, access to the symbolism *is no longer restricted to a privileged minority*. In order to elaborate on the points, we will need just a little more historical perspective, before we focus our attention on the digital media themselves. As recounted in Shaffer and Kaput (1999), the development of computational media required three elements: the existence of discrete notations without fixed reference fields (that is, the idea of formalism), the creation of syntactically coherent rules of transformation on such notations, and a physical medium in which to instantiate these transformations outside the human cortex and apart from human physical actions. Hence in the 20th century a profound shift has occurred, from operable notation systems requiring a suitably trained human partner for execution of the operations, to systems that run autonomously of a human partner.

ON CHANGING REPRESENTATIONAL INFRASTRUCTURES

Our starting point is the assertion that the extent to which a medium becomes infrastructural is the extent to which it passes as unnoticed. This is fine, until one needs to be aware of the structural facets of the medium, in order to learn either how to express oneself within it, or understand what might be expressed within it (or both). From the point of view of the learner, this can be confusing. For example, Leibniz' notation is a 2nd order notation built on top of algebra because it guides actions on expressions built in algebraic notation (a fact that confuses many students even today who do not distinguish between "taking the derivative" and simplifying the result—after all, they are both ways of transforming strings of symbols into new strings of symbols). Since the ideas, constructions, and techniques of Calculus are written in the language of algebra, knowledge of Calculus has historically depended on knowledge of algebra. This in turn means that this knowledge has historically been the province of a small intellectual elite—despite the fact that the key underlying ideas concerning rates, accumulations, and the relations between them are far more general than the narrow algebraic representations of them in most curricula (Kaput, 1994).

Representational forms are often transparent to the expert. Musicians do not "think" about musical notation as they play an instrument any more than expert mathematicians have to (except when they are constructing a new notation or definition). But when one is learning or constructing something new, one needs to think explicitly about the representational system itself—we require, in other words, that the representational system is simultaneously transparent and opaque. This "coordinated transparency" (Hancock, 1995) represents a synthesis of meaning and mechanism, a situation (desirable but not always easily achievable) in which fluency *with and within* the medium can temporarily be replaced by a conscious awareness of its (usually invisible) internal structures. Grammar checking (human rather than computational) is a good example.

As noted above, the development of algebraic representational forms which generated fluency among the cognoscenti, took place within the semiotic constraints of static, inert media, and *largely without regard to learnability outside the community of intellectual elite involved*. Over the past several centuries, this community's intellectual tools, methods, and products (the foundations of the science and technology upon which we depend) were not only institutionalized as the structure and core content of school and university curricula in most industrialized countries and taken as the epistemological essence of mathematics (Bochner, 1966; Mahoney, 1980), but in most countries became the yardstick against which academic success was defined. Thus the close relationship of knowledge and its culturally-shared preferred representations, precisely the coupling that has produced such a powerful synergy for developing scientific ideas since the Renaissance, became an obstacle to learning, and even a barrier which prevented whole classes from accessing the ideas which the representations were so finely tuned to express.

While the execution of processes was necessarily subsumed within the individual mind, decoupling knowledge from its preferred representation was difficult.[3] But as we have seen, this situation has now changed. The emergence of a virtual culture has had far-reaching implications for what it is that people need to know, as well as how they can express that knowledge. We may, in fact, have to reevaluate what knowledge itself is, now that knowledge *and the means to act on it* can reside inside circuits that are fired by electrons rather than neurons. Key among these implications is the recognition that algorithms, and their instantiation in computer programs, are now a ubiquitous form of knowledge, and that they—or at least the outcomes of their execution—are fundamental to the working and recreational experiences of all individuals within the developed world.

Many individuals and social groups have suffered a massive deskilling of their working lives precisely because of this devolution of executive power to the machine. But not all. Indeed many occupations (or at least parts of them) have become more challenging and enriching because of the introduction of digital technologies. What is common, however, is that the relationship between computational systems and individuals has become much more intimate than was ever envisaged. In part, this is a simple maturation of the technology—three of the more obvious and striking aspects are its miniaturization, the power of graphical displays and, of course, its connectivity (in 1982, the idea of communication as a central functionality of computers was the preserve of only a few experts in universities).

These technical facets of computer systems and the ways we use them, have reshaped our relationship with them. On the one hand, they have reduced even further the necessity for "users" to make sense of how computational systems do what they do. The intimacy that, for example, a painter has with her brush—or perhaps much more relevant analogy, the relationship between a musician and her instrument—is rarely (currently) possible with the computer, despite the close proximity and personal relationship which many people have with their machines, especially hand-held ones.

An accepted (but, as we shall see, fundamentally false) pedagogical corollary is that since mathematics is now performed by the computer, there is no need for "users" to know any mathematics themselves (for a well-publicized but disappointing set of arguments propound-

ing this belief, see Brammall and White, 2000). Like most conventional wisdoms, this argument contains a grain (but only a grain) of truth. Purely computational abilities beyond the trivial, for example, are increasingly anachronistic. Low-level programming is increasingly redundant for users, as the tools available for configuring systems become increasingly high-level. Taken together, one might be forgiven for believing that the devolution of executive power to the computer removes the necessity for human expression altogether (or at least, for all but those who program them).

In one sense this is true. Precomputational infrastructures certainly make it necessary that individuals pay attention to calculation: and generations of "successful" students can testify to the fact that calculational ability can be sufficient (e.g., for passing examinations) even at the expense of understanding how the symbols work. In fact, quite generally, the need to think creatively about representational forms arises less obviously in settings where things work transparently (cogs, levers, pulleys have their own phenomenology). Now the devolution of processing power to the computer has generated the need for a new intellectual infrastructure; people need to represent for themselves how things work, what makes systems fail and what would be needed to correct them. This kind of knowledge is increasingly important; it is knowledge that potentially unlocks the mathematics that is wrapped invisibly into the systems we now use, and yet understand so little of. Increasingly, we need—to put it bluntly—to make sense of mechanism.

Yet the need to make sense of mechanism is not fundamentally new. Indeed, the syntax of the numerical, algebraic and calculus representation systems can be regarded as mechanisms, and the bulk of mathematics schooling has been devoted to teaching and learning that form of mechanism. However, there is a further complexity to the present situation. It is true that fewer and fewer people need to program computers, at least in the usual sense of the term "program.". But more and more people need to know something of how the machines and the systems (social, professional, financial, physical) operate—*not* just the few who are responsible for building them. We cannot adduce evidence for this assertion here (for a convincing selection of papers on this theme, see Hoyles, Morgan, & Woodhouse, 1999).

We share a vision of a mathematics curriculum that assumes mathematical understanding should be built around the construction and interpretation of quantitative and semi-quantitative models, where students explore mathematical technologies and analyze methods in contexts that show how they can be used and why they work in the way they do. We can also refer the interested reader to a series of papers, which have studied the mathematics of professional practices in a number of areas (aviation pilots, nurses, bank employees, and most recently, engineers; see, for example, Noss & Hoyles,1996a; Pozzi, Noss, & Hoyles, 1998; Noss, Pozzi, & Hoyles, 1999; Hoyles, Noss, & Pozzi., 2001).

We restrict ourselves to two observations. First, at critical moments of their professional practice, people try to make sense out of complex situations by building mental models, or, if they do not have access to the raw material of model-building, by circumventing them. Circumvention (ignoring inconvenient data) can be a dangerous strategy. To gain access to underlying models, to make them *visible*, is to focus on the quantities that matter, and on the relationships among them. In order to gain such a sense of mechanism, one needs interpretative knowledge about, for example, graphs, about parameters and variables, about continuity, and a broad range of representational abilities that are different from, but no less important than, calculational and manipulational skills we have inculcated in young people until now.

The second observation concerns the complexity of interaction between professional and mathematical practices. It is true that more and more professional practice devolves calculational expertise to the computer. But it is not true (or if it is, it is dangerously so) that the computers can be left to make judgment (one of our examples concerns a life-and-death decision on a pediatric ward—see Noss, 1998). Judgment in the presence of intimate computational power requires new kinds of representational knowledge: distinguishing between what the computer is and is not doing; what can be easily modified in the model and what

cannot; what has been incorporated into the model and why; and what *kinds* of model have been instantiated. As examples, we may consider the difference between parallel and serial computational models, how different kinds of knowledge are encoded with them, and what kinds of interpretation they allow; and, not least, the *communicative* value of representational knowledge, in terms of sharing knowledge with others who interact with other parts of the same system, or other, linked systems.

The new element in the situation is, of course, that the systems that control our lives are now built on mathematical principles. This is a major—perhaps *the* major—property of the virtual culture. The devolution of execution to the machines means more than this: not only do the machines now *do* mathematical execution, it implies that any consequential appreciation of what the machines do must itself be based on mathematical principles. If an individual does not have the means formally to relate his or her intellectual model of the mathematical principles with those inside the machine, then appreciation of the model must necessarily be partial.

Of course, this does not mean that such models need to be expressed in the same languages as used inside the machine. Quite the reverse. It means that we have to find ways to help people to capture the dynamics of the system, so they can follow the consequences of particular actions while maintaining a realistic sense of the structures of relationships between them. We now turn to some examples which begin to address the issues raised, and then show how students can be stimulated to explore mathematical mechanisms and in so doing rebuild the synergy of knowledge and representation.[4]

A NEW REPRESENTATIONAL INFRASTRUCTURE FOR CARTESIAN GRAPHS COUPLED WITH EMBEDDED DERIVATIVES AND INTEGRALS LINKED TO PHENOMENA

Over the past two decades, the character string approaches to the mathematics of change and variation have been extended to include and to link to tabular and graphical approaches, yielding the "the Big Three" representation systems, algebra, tables, and graphs frequently advocated in mathematics education. However, almost all functions in school mathematics continue to be defined and identified as character-string algebraic objects, especially as closed form definitions of functions—built into the technology via keyboard hardware. In the SimCalc Project, we have identified five representational innovations, all of which require a computational medium for their realization but which do not require the algebraic infrastructure for their use and comprehension. The aim in introducing these facilities is to put phenomena at the center of the representation experience, so children can see the results, in observable phenomena, of their actions on representations of the phenomenon, and vice versa. These are:

- The definition and direct manipulation of *graphically defined and editable* functions, especially piecewise-defined functions, with or without algebraic descriptions. Included is "snap-to-grid" control, whereby the allowed values can be constrained as needed (to integers, for example) allowing a new balance between complexity and computational tractability. This facility means that students can model interesting change situations while avoiding degeneracy of constant rates of change, and postponing (but not ignoring!) the messiness and conceptual challenges of continuous change.
- *Direct, hot-linked connections* between functions and their derivatives or integrals. Traditionally, connections between descriptions of rates of change (e.g., velocities) and accumulations (positions) are mediated through the algebraic symbol system as sequential procedures employing derivative and integral formulas, which is the main reason that calculus sits at the end of a long sequence of curricular prerequisites

- Direct connections between these new representations and simulations to allow immediate construction and execution of variation phenomena.
- Importing physical motion-data via Micro Computer Based Lab or Calculator Based Lab (MBL/CBL) and reenacting it in simulations, and exporting function-generated data to define Line Becomes Motion (LBM) to drive physical phenomena (including cars on tracks).

We also employ hybrid physical/cybernetic devices embodying dynamical systems, whose inner workings are visible and open to examination and control, and whose quantitative behavior is symbolized with real-time graphs generated on a computer screen.

We will risk real danger by providing grayscale snapshots of colorful, dynamic, interactive lessons, especially by superimposing multiple problems and solutions on the same graphs. We will provide some basic activities to illustrate concretely, albeit thinly, how this new representational infrastructure can work. First note that the various graphs appearing in the figures below are created piecewise simply by clicking, dragging and/or stretching segments, although in other activities it is also possible to specify the graphs algebraically, or by importing data, or by (partially constrained) drawing. (A similar set appears in Kaput, 2000.)

Variation, area, average, approximation, slope, continuity, and smoothness for jerky elevators

Suppose we are given a staircase velocity function (see Figure 26.1), which drives the motion of the left-hand elevator to its left (these are color-coded in the software). The following kinds of lesson snippets are usually preceded by context-rich work that involves moving elevators around to accomplish various tasks, such as delivering pizzas to various floors, etc.

1. How will the elevator move if driven by Plot #1 (the piece-wise downward staircase velocity function in Figure 26.1), and where will it end its trip? (It starts on the 0th floor.)
2. Does there exist a constant velocity function for the 2nd elevator (just to the right of the 1st) that gets to the same final floor at exactly the same time as the 1st? If so, build it. (Plot #2—the one-piece constant velocity function)
3. Make a linear velocity function for the 3rd elevator that provides a smoothly decreasing velocity approximating the motion of the 1st (staircase) elevator. Before running it, predict how far apart the 1st and 3rd elevators will finish their trips. (Plot #3—the linear decreasing velocity function)

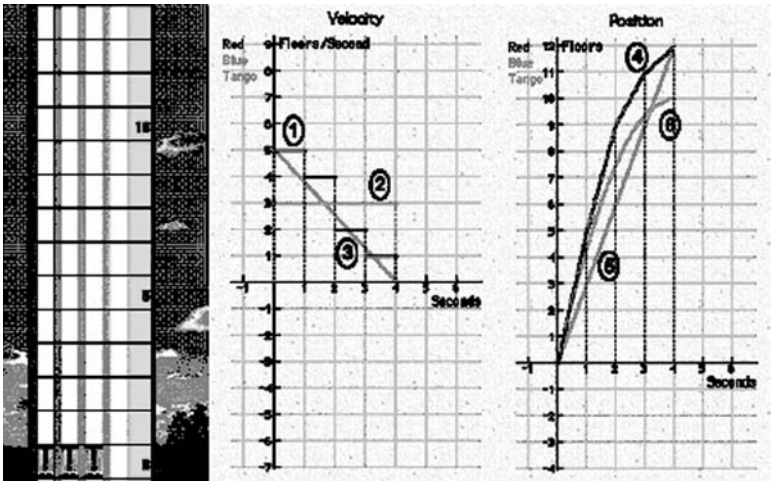

Figure 26.1 A staircase velocity function.

4. For the staircase velocity function (Plot #1) in Figure 26.1, what is the corresponding position graph, and what is its slope at 3.5 seconds? (Plot #4—the piece-wise increasing position function)
5. While the average-velocity function for the staircase has exactly the same area under it as the staircase, what is an easy way to draw its corresponding position graph? (Plot #5—the linear increasing position function)
6. What is the key difference between the position graph for the staircase and the position graph of the 3rd velocity function? (Plot #6—the quadratic position function)

Question 1 involves interpreting variation via a variable velocity function whose integral, to determine the final position, can be determined with whole number arithmetic. Such step-wise varying rate-functions are intensively used in SimCalc instructional materials to build the notion of area as accumulation. Such functions also raise issues of continuity, acceleration, and physical realizability of simulations that are explored in depth using MBL and LBM technologies. Of course, they also occur in economic situations with great frequency—tax rates, pay rates, telephone rates, etc.

Question 2 introduces the key idea of average, which, via our ability to use snap-to-grid to control the available number system also enables examination of when the average "exists" and whether it must inevitably equal the value of the varying function at some point in its domain. In traditional instruction, most students only experience continuously changing rates and hence never really confront the issue since the average always hits the continuously varying intermediate values.

Question 3 points up the reversal of the usual relationship between step functions and continuous ones (usually, the former are used to approximate the latter), and highlights the integration of fraction and signed number arithmetic in the Mathematics of Change and Variation Curriculum (MCV). Position graphs and linearly changing velocity are developed over many lessons in many ways, including the differences between physical motion (including force-acceleration issues), and economic functions, so the glimpse here may be misleading in its abruptness.

Question 4 introduces the idea of slope as height of velocity segment, and is part of an extensive study of slope as rate of change.

Question 5 illustrates the power of a "2nd opinion" since the position description of the average velocity motion is merely a straight line joining the start and end of the position graph. These two ways of describing change-phenomena are treated as complementary throughout SimCalc instructional materials.

Question 6 deals with smoothness, and is part of an extended introduction to quadratic functions as accumulations of linear ones that weaves back to issues of acceleration and physical motion, and their physical realizability. An accompanying set of investigations examine non-physical motion, e.g., price or other money rates that change discontinuously such as tax rates, phone rates, royalty rates, etc.

Activities linking velocity and position descriptions of motion in the context of signed numbers and areas

The earlier parts of the next lesson, from which these snippets are taken, involve students in creating graphs to move Clown and Dude around, switching places at constant speed, coming together and then returning to their original positions, and so on. (Only step-wise constant velocities have been made available here, although other function types could have been used, and, in fact, are used in SimCalc materials.)

Challenge: Clown and Dude are to switch their positions so that they pass by each other to the left of the midpoint between them and stop at exactly the same time. First, after marking off a line about 12-feet long, you and a classmate walk their motions!

> Now make a *position* graph for Clown and a *velocity* graph for Dude so that *they* can do this.

The student needs to construct graphs similar to Plot #1 (on the position graph) and Plot #2 (on the velocity graph) in Figure 26.2. We have also shown the respective corresponding velocity and position graphs, Plots #3 and #4, which can be revealed and discussed later. Note that velocity and position graphs are hot-linked, so changes in the height of a velocity segment are immediately reflected in the slope of the corresponding position segment, and vice-versa. Importantly, the activity requires interpretations of positive and negative velocities, and hence provides meaningful work with signed number arithmetic, as well as the representation of simultaneous position—paving the way for simultaneous equations. Later activities involve a story-line where Dude is patrolling the area (periodic motion) and Clown gets "interested" in Dude, follows him at a fixed distance, "harasses" him, and eventually, they dance—where the student, of course, is responsible for making the dance.

Extensions to MBL and LBM

The above representational innovations can be combined with the principals illustrated by Questions #5 & #6, mentioned earlier, to create opportunities to study the Math of Change & Variation. For example, we can import and display motion data in the classic MBL/CBL ways, but in addition, we can now attach this physically-based data to the objects in a simulation and replay their motion, compare it with motions defined synthetically, so that a student can perform and import a physical motion that can lead an entire group of dancers whose motions are created synthetically. Further, a student can define a motion using a mathematical function (position or velocity) in any of the ways one might care to define a function and then "run" it physically in a linked LBM miniature car on a track. The forms of learning supported by these kinds of devices and activities, especially how they relate to one another and to physical intuition, are under active investigation, and the study of this richly populated space of interrelated inscriptions, and the new connections among physical, kinesthetic, cybernetic, and notational phenomena, will continue for years to come—and will be instantiated in increasingly networked contexts (Kaput, 2000; Nemirovsky, Kaput, & Roschelle, 1998).

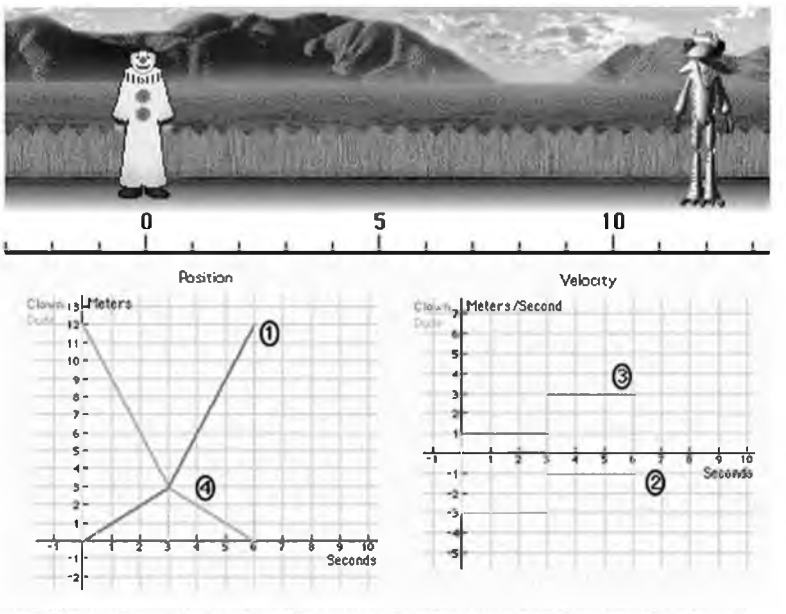

Figure 26.2 You, and then Clown & Dude, cross to the left of the center.

The result of using these systems, particularly in combination and over an extended period of time, is a qualitative transformation in the mathematical experience of change and variation. However, short term, in less than a minute, using either rate or totals descriptions of the quantities involved, or even a mix of them, a student as early as 6th–8th grade can construct and examine a variety of interesting change phenomena that relate to direct experience of daily phenomena. And in more extended investigations, newly intimate connections among physical, linguistic, kinesthetic, cognitive, and symbolic experience become possible.

PRELIMINARY REFLECTIONS ON THE NEW REPRESENTATIONAL INFRASTRUCTURE

A key aspect of the above representational infrastructure is revealed when we compare how the knowledge and skill embodied in the system relates to the knowledge and skill embodied in the usual curriculum leading to and including Calculus. At the heart of the Calculus is the Fundamental Theorem of Calculus, the bidirectional relationship between the rate of change and the accumulation of varying quantities. *This core relationship is built into the infrastructure at the ground level.* Recall that the hierarchical placeholder representation system for arithmetic and the rules built upon it embody an enormously efficient structure for representing quantities (especially when extended to rational numbers) which in turn supports an extremely efficient calculation system for use by those who master the rules built upon it. This is true of the highly refined algebraic system as well. Similarly, this new system embodies the enormously powerful idea of the Fundamental Theorem in an extremely efficient graphically manipulable structure that confers upon those who master it an extraordinary ability to relate rates of change of variable quantities and their accumulation. In a deep sense, the new system amounts to the same kind of consolidation into a manipulable representational infrastructure an important set of achievements of the prior culture that occurred with arithmetic and algebra.

DEVELOPING A SENSE OF MECHANISM

In this penultimate section, we focus on a corpus of work which is emerging from the *Playground* project (see http://www.ioe.ac.uk/playground). Like the SimCalc examples above, our interest focuses on new ways to express mathematical relationships, bringing children into contact with mathematized descriptions of their realities at ages much younger than we would normally countenance with static technologies. Unlike the SimCalc example, which typically involves students beginning at ages 11–12 (although it is also used at the university level), we are trying to explore what might be gained by very young children (aged between 4 and 8) building their own executable representations of relationships—in effect, we are redefining the idea of programming.

The rationale for programming has a long and distinguished history, stretching back some 30 years or more (see, for example, Feurzeig, Papert, et al., 1969). We have no intention of rehearsing the argument here (see Noss & Hoyles, 1996b, for a history and rationale for programming in the context of mathematical learning). What is new is that programming has begun to change its character, having been expressed in various forms—as text (still the dominant form) as icons, and now, as we shall see, as animated code. We believe that this last change of expressive form marks a significant shift in what is possible for young children.

Our central focus is to open possibilities for children to design, construct and share their own video games. We are designing computational environments for children to build and modify games using the formalization of rules as creative tools in the constructive process. We call these environments "Playgrounds." We are working with two new and evolving

programming systems, *ToonTalk*—an animated programming language (Kahn, 1999)—and *Imagine* a concurrent object-oriented variant of the Logo programming language (Blaho, Kalas, & Tomesányi, 1999: note, at this point, the language was named "OpenLogo"). Each of our two Playgrounds represents a layer we have built on top of these platforms, incorporating elements that allow multiple entry points into the ideas of formalizing rules. In this paper we concentrate on our work with *ToonTalk*.

Our objective has a strong epistemological rationale. The challenge is to find ways for very young children to use non-textual means to express and explore the knowledge which underpins the genre of video games: what it means for objects to collide, how 2-dimensional motion of an object (or a mouse, or a joystick) can be thought about, the construction of animation, and the hundreds of little pieces of knowledge which go to make up the workings of video games. We see this as an instantiation of a much broader class of knowledge, which, quite simply, we call developing a sense of mechanism.

Our choice of video games builds on established work by, for example, Kafai (1995) in that it has chosen a domain which seems to be naturally attractive for many children. We have no ulterior pedagogical or epistemological motive: we do not ask the children with whom we work to design games for any purpose other than their own amusement. Testing our intuitive belief that games themselves form a sufficiently-rich backdrop against which to explore mathematical relationships forms part of our studies with children, and forms a central element of our design brief.

ToonTalk is a world in which animations themselves are the source code of the language; that is, programs are created by directly manipulating animated characters, and programming is *by example* (see Cypher, 1993). A full description can be found in Kahn, 1999). *ToonTalk* is constructed around the metaphor of a city, populated by houses (in which programs or methods are built), trucks are dispatched to build new houses (new processes spawned), robots are trained (for new programs or methods), and birds fly to their nests (message passing). A helicopters allows the user to navigate around the city, or to hover above it watching trucks move around (as an aside, and to emphasize that *ToonTalk* is a Turing equivalent language, it is both instructive and surprising to watch a city recursively grow and shrink as a quicksort is being executed).

Robots are trained to carry out tasks inside houses (defining the body of a method). A user trains a robot by entering its "thought bubble" and controlling it to work on concrete values (see Figure 26.3). The robot remembers the actions in a manner that can easily be

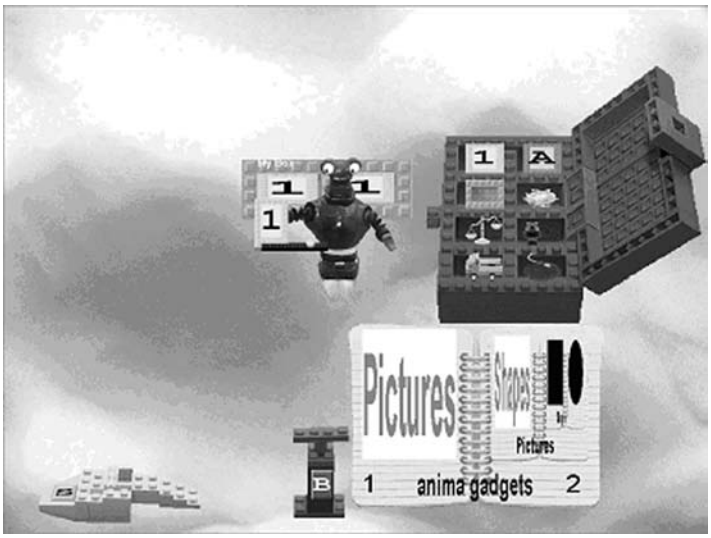

Figure 26.3 A robot is trained to add one value to another.

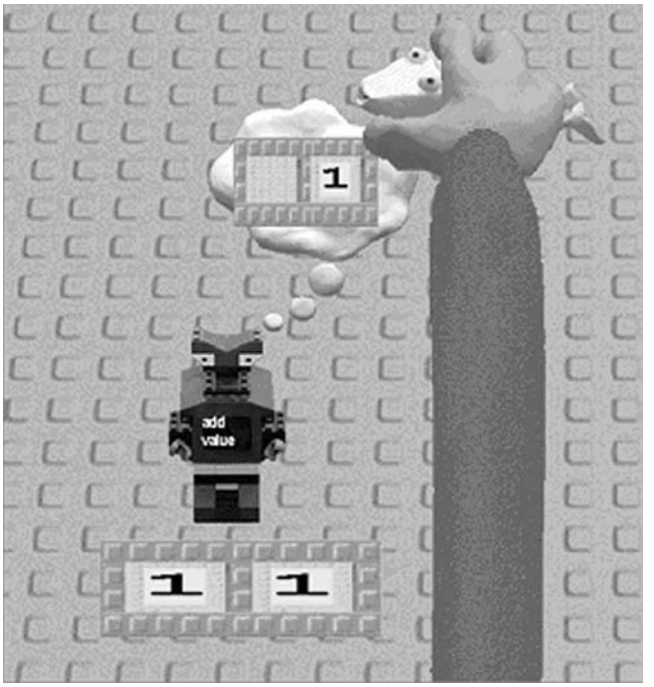

Figure 26.4 Relaxing constraints by removing details.

abstracted to apply in other contexts by later removing detail from the robot's thought bubble (see Figure 26.4). Message passing between methods (robots) is represented as a bird taking a message to her nest, and changing a tuple is achieved by taking items out of compartments of a box and dropping in new ones.

It is quite difficult in text and static graphics to convey the feel of programming with *ToonTalk*. The metaphor of moving around in a city is pervasive, and the sense of object-oriented programming in an environment by direct manipulation, is a novel experience for those of us who believed that symbolic formalism of programming made interaction on a textual level inevitable.

The nature of the platform is paramount. Our choice of *ToonTalk* implied that any layers we built above it had to mesh with the metaphors of the platform. Our aim was to design a permeable abstraction barrier between ready made pieces of open code with multi-modal representations (we call these "behaviors": some examples will be given below) and the *ToonTalk* language itself lying underneath. This stands in marked contrast to some modern programming languages such as Java and C++ which by default enforce these abstraction barriers and do not allow programmers using predefined objects to discover their underlying implementation. But the crucial dimension, which dictated the design of the playground layer, was that of openness. At any level of granularity, an element should be decomposable into smaller pieces down to the lowest level of the animated *ToonTalk* programming language. Indeed, as we began to see children decomposing the games and sharing their parts across sites and countries, it became clearer that we were working in a design paradigm akin to component software architecture (CSA). While some (but not all) of the component community are concerned to a greater or lesser extent with the adaptability of their components, for us it central. We are concerned with designing software for investigating mechanism; individual components therefore need to have intuitive windows to their workings and a means for modification. (For more information on the role of behaviors in playgrounds, see Hoyles, Noss, & Adamson, 2002).

In the design of an environment where the opening of mechanisms is the primary objective, it is not only desirable that pieces are easily opened but that they afford access to their workings through an intuitive interface. In traditional CSA, the user interacts with the interface model provided by the architecture but not the implementation of individual components. In our open component model, we require both. Users should be able to work at several levels simultaneously; (a) composing components where necessary as wholes relying solely on the interface for component manipulation, and (b) opening a component to reveal the source code whenever modification or inspection of the component is desired. To facilitate this, we need to ensure that components interoperate at a technical level but also that manipulation at interface and implementation levels is made intuitive by a high degree of *semantic* interoperability. In other words, if users are to use, share and manipulate components in the construction of larger pieces of software, consistency of interface and multiple ways of accessing the functionality become important criteria in their design.

An illustration: The space behaviors game

We start with a game based loosely on the space invaders genre of shooting games. The player controls a space ship that can fire white bullets in four directions. If a white bullet hits an invader, it blows up, but if an invader hits the spaceship, the spaceship is destroyed. The aim is to destroy the three invaders before they hit you.

The two boys in our case study wanted to change the appearance of the spaceship. Working at the surface level of changing the appearance of objects is relatively simple. Objects and behaviors are interoperable, so they simply had to select their new object and transfer all the behaviors across by placing the old object on the back of the new one. They chose a Pokémon character called Pikachu as their new representation of the space ship. Pikachu is associated with lightning so the boys wanted it to shoot bolts of lightning rather than white bullets. Turning over Pikachu to reveal its robots and behaviors, they could immediately identify the firing mechanism through its visual representation (see Figure 26.6).

The specific representation here is quite subtle, as the actual white bullets are not immediately visible. A modular visually represented architecture is required to give the clues for further inspection (see Figure 26.7)

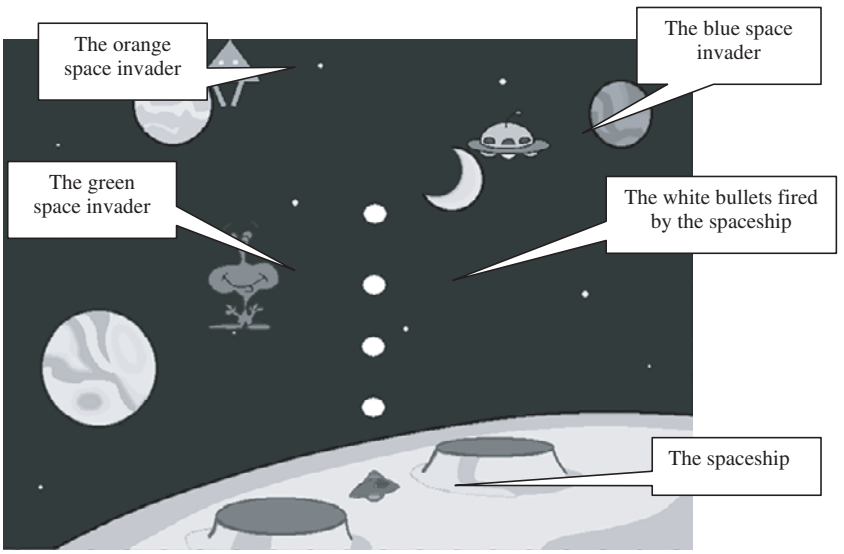

Figure 26.5 The original 'space behaviors' game.

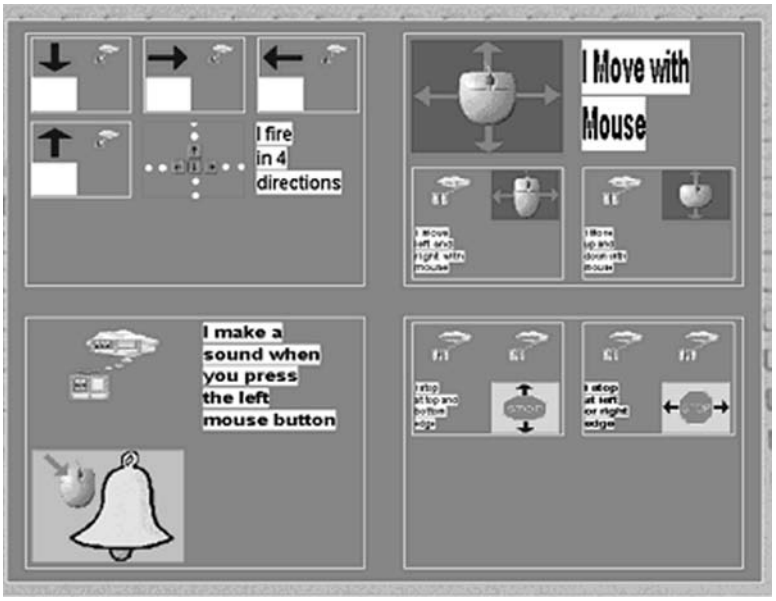

Figure 26.6 The behaviors on the spaceship.

At each level, the level below is visible. In Figure 26.7, the firing behavior is shown is successive stages of exposure. Taking apart the behavior down to the lowest level reveals the white bullet. We are hardly in a position to claim that the boys fully understood the meaning of all the inputs to the firing robot or how the robot actually worked, but they could simply home in the bullet. They knew that its functionality had to be taken over by lightning. The boys removed the bullet picture and put it on the back of a lightning picture and did this four times—one for each direction. Having modified the input to these four behaviors in this way and put them all back together on the back of Pikachu, the boys were ready to try out their modified game.

But would it work? The boys thought so, but, as it turned out, they were wrong. The changes to the spaceship worked as expected and Pikachu could fire lightning in four directions, but now the boys noticed that the invaders appeared to be indestructible. They were not sure why and guessed that it was because the destroy behavior had somehow gone missing from the back of the invaders. They decided to check this out and took off the blue invader from the scene and turned it over to investigate its behaviors.

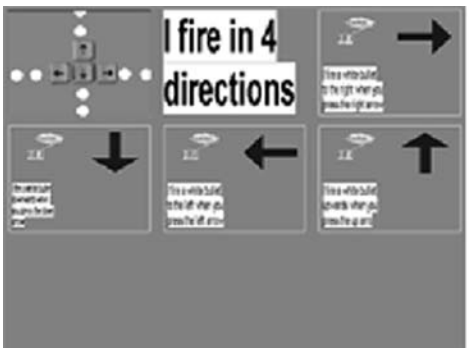

Figure 26.7a The complete firing mechanism with four firing components.

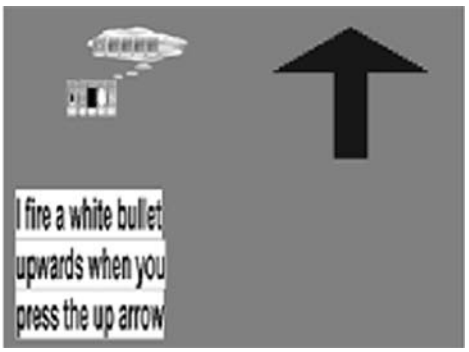

Figure 26.7b The firing up component only.

Figure 26.7 The firing behavior shown is successive stages of exposure.

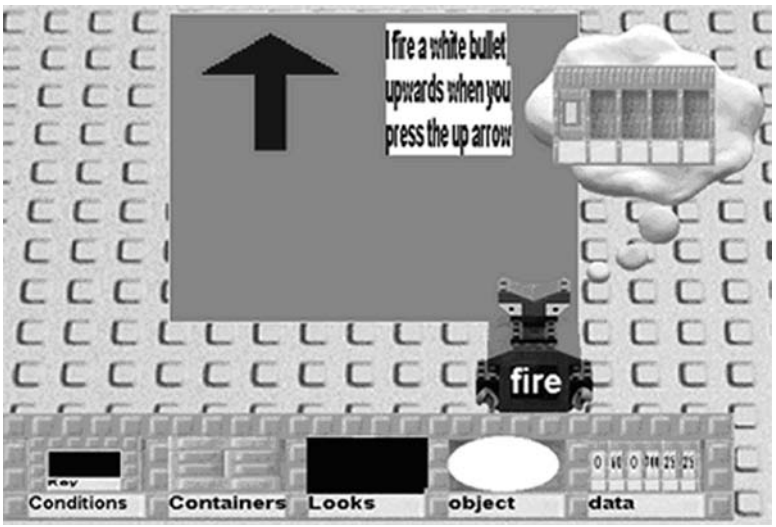

Figure 26.8 Successive revelation of the *ToonTalk* code.

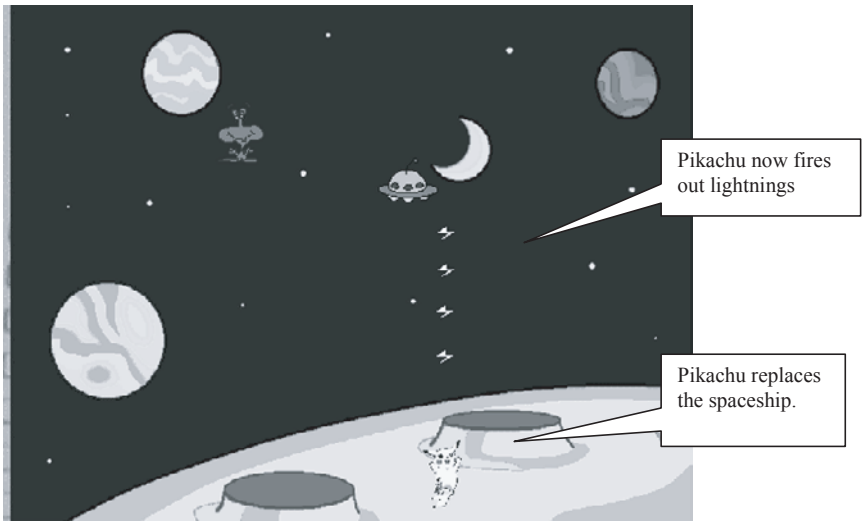

Figure 26.9 The game after changes made at the surface level and input level.

They recognized the relevant behavior by first noticing the explosion icon and the written rule, "I disappear when I touch a white bullet." They removed this behavior from the back of the invader so they could study the next level and look at the actual code (see Fig. 26.11). They would now "see" the concrete representation of the condition for the robot, i.e., that it would perform its action (destroy object) when hit by a white bullet.

Seeing how the rule worked helped the boys debug what had gone wrong: the lightning picture had to appear in the place of the white bullet. Again, without having to appreciate how all the pieces of the mechanism worked, the boys could make this replacement and so achieve their rule change (see Figure 26.11 and Figure 26.12).

It might be that some readers will be wondering what this has to do with mathematics. Our reply is that it is about rules expressed formally and their implications, and that—as we have argued earlier—this is a central aspect of what it means to think mathematically in the computational era. At a more detailed level, this claim breaks down into two subclaims. First,

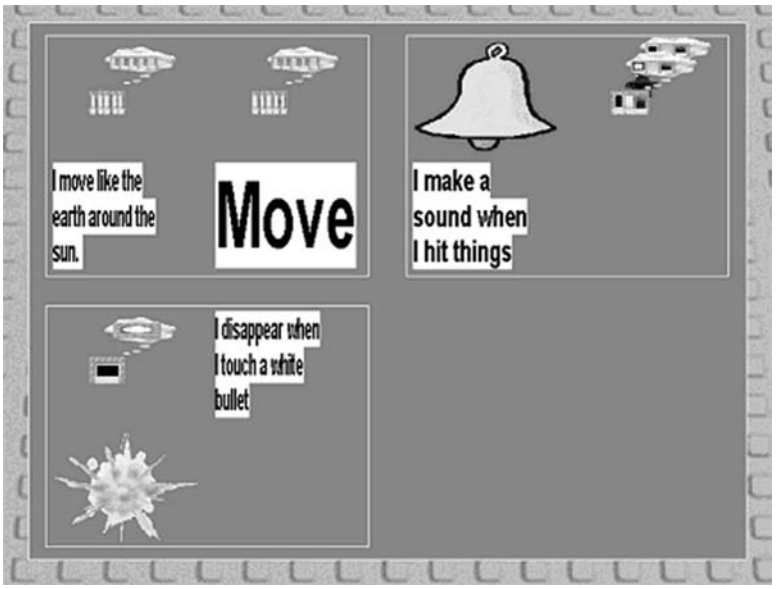

Figure 26.10 The back of the blue invader.

it is about children learning that there *are* rules that have implications for what is modeled and what is observed and these rules are something over which they have some control. Second, these rules embody and are built upon a previously-constructed representational infrastructure that offers extraordinary power to those who master it. Over the 2 years of the Playground project, we have collected numerous examples of children taking apart a scene and exploring how it worked, why it worked and how it could be changed. Often, a teacher was involved: in fact, a key aspect of the claim is that such an approach was more teachable than other programming environments, as the things that mattered are visible and easily manipulable and the granularity of the pieces customized by the teacher for the learner. As with the SimCalc infrastructure, increasing learnability and expressive power for all students are fundamental goals.

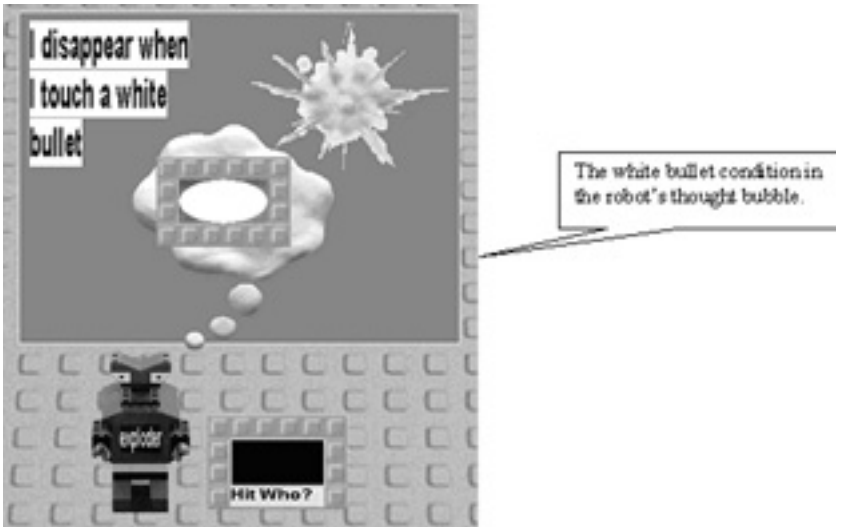

Figure 26.11 The exposed *ToonTalk* code showing the white bullet condition.

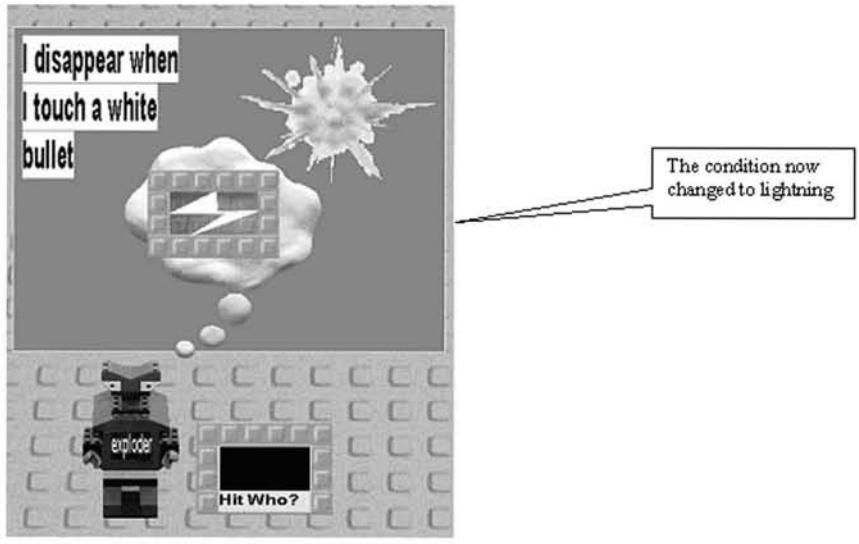

Figure 26.12 The code after changing the condition to lightning.

CONCLUSIONS

In this chapter, we have attempted to show how the evolution of representational infrastructures and associated artifacts and technologies have, over long periods of time, gradually externalized aspects of knowledge and transformational skill that previously existed only in the minds and practices of a privileged elite. We have sought to show how changes in representational infrastructure are intimately linked to learnability and the democratization of intellectual power. We have illustrated this point by reference to the development of number and algebraic notation, calculus, SimCalc graphs, and ideas of open, manipulable mechanism. In each example, the physical instantiation of these notations directly enlarged the limited processing power of human minds as well as affording experience of new domains of knowledge to solve new problems among populations who previously had no access to that knowledge and intellectual capacity. Computational media have provided a next step in the evolution of powerful, expressive systems for mathematics.

We have endeavored to illustrate our major contention: that mathematicians and mathematics educators need to turn their attention to defining these newly empowering representational infrastructures for children. In the past, beginning with writing itself (Kaput, 2000), more powerful representational infrastructures have been a source of intellectual and mathematical power, but at a cost of learnability and hence access. Hence they tended to remain the province of an elite minority who were inducted into their use. New computational media offer the opportunity to create democratizing infrastructures which will redefine school knowledge (for a fuller discussion of these issues, see Noss & Hoyles, 1996b). Viewed optimistically, these will exploit the processing power of the new media while at the same time ensuring that students maintain an intuitive feel of the central knowledge elements at work and how they relate to each other. Yet, if the power and potential of computers is to be exploited in school mathematics, attention must be paid to this level of representational infrastructure. A companion need is to develop sustained curricula and modes of teaching and learning that incorporate and exploit these new representations and that encourage students to develop their meta-representational abilities (diSessa, 2000) so they become fluent with new systems of expression as they arise, can create and modify such systems themselves, and can make wise choices among them as these systems proliferate in the coming decades.

Thus we wish to challenge our community to focus attention on the design and use of representational infrastructures that intimately link to students' personal experience. This is a necessary step if we are to move away from a 19th-century school of mathematics concentrating on isolated skills based on static representational systems in a tightly-defined curriculum (with only a minority able to engage in independent problem solving). Our contention is that knowledge produced in static, inert media can become learnable in new ways, and new representational infrastructures and systems of knowledge become possible, serving both the learnability of previously constructed knowledge and the construction of new knowledge.

NOTES

1. In fact, there was some resistance to writing abacus results in Hindu-Arabic numerals on the grounds that they could be easily altered, e.g., changing a "6" to an "8" by adding a mark to the top part of the "6," or a "9" could be made from a "1," and so on.
2. As a way of computing (derived, of course, from the Roman word for pebble, since pebbles were used for computation, ironically, because the Roman numeral system was so inefficient).
3. But not impossible: indeed, a considerable amount of mathematical education research has tried to study—and encourage—the ways in which people form conceptual images of mathematical ideas independently of—and sometimes in conflict with—the preferred algebraic or formal representation.
4. The former provides a putative enhancement of experiential phenomena, and thus a richer base for intuitive knowledge. This is hardly unique to digital technologies: when mechanized transport was first invented, people for the first time, found it obvious that centrifugal force was something to do with changing direction (the fact that it feels like centrifugal force rather than centripetal acceleration just shows that intuitions don't always give the whole picture).

REFERENCES

Blaho, A., Kalas, I., & Tomcsányi, P. (1999). OpenLogo—a new implementation of Logo. In R. Nikolov, E. Sendova, I. Nikolova, & I. Derzhanski. (Eds.), *Proceedings of the Seventh European Logo Conference,* 95–102.

Bochner, S. (1966). *The role of mathematics in the rise of science.* Princeton, NJ: Princeton University Press.

Boyer, C. (1959). *The history of calculus and its historical development.* New York: Dover.

Brammall, S., & White, J. (2000). *Why learn maths?* London: Bedford Way Papers, Institute of Education.

Bruner, J. (1973). *Beyond the information given.* New York: Norton.

Clement, J. (1982). Algebra word problem solutions: Thought processes underlying a common misconception. *Journal for Research in Mathematics Education, 13*(1), 16–30.

Cypher, A. (Ed.). (1993). *Watch what I do: Programming by demonstration.* Cambridge, MA: MIT Press.

Dantzig, T. (1954). *Number: The language of science.* New York: Macmillan.

diSessa, A. (2000). *Changing minds, computers, learning and literacy.* Cambridge, MA: MIT Press.

Donald, M. (1991). *Origins of the modern mind: Three stages in the evolution of culture and cognition.* Cambridge, MA: Harvard University Press.

Edwards, C. (1979). *The historical development of the calculus.* New York: Springer-Verlag.

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1969). *Programming languages as a conceptual framework for teaching mathematics.* Report No. 1889. Cambridge, MA: Bolt, Beranek and Newman.

Hancock, C. (1995). The medium and the curriculum: reflections on transparent tools and tacid mathematics. In diSessa, A. A., Hoyles, C., & Noss, R. (Eds.), *Computers and exploratory learning* (pp. 221–241). Berlin: Springer-Verlag.

Hoyles, C., Morgan, C., & Woodhouse, G. (Eds.). (1999). *Rethinking the mathematics curriculum.* London: Falmer Press.

Hoyles C., Noss R., & Pozzi S. (2001). Proportional reasoning in nursing practice *Journal for Research in Mathematics Education, 32*(1), 42, 4–27.

Hoyles, C., Noss, R., & Adamson, R. (2002). Rethinking the microworld idea. *Journal of Education Computing Research, 27*(1&2), 29–53.

Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning.* Mahwah, NJ: Erlbaum.

Kahn, K. (1999). Helping children learn hard things: computer programming with familiar objects and activities. In Druin, A. (Ed.), *The design of children's technology* (pp. 223–241). San Francisco: Morgan Kaufman.

Kaput, J. (1994). The representational roles of technology in connecting mathematics with authentic experience. In R. Bieler, R. W. Scholz, R. Strasser, & B. Winkelman (Eds.), *Mathematics didactics as a scientific discipline* (pp. 379–397). Dordecht, The Netherlands: Kluwer.

Kaput, J. (2000). Implications of the shift from isolated, expensive technology to connected, inexpensive, ubiquitous, and diverse technologies. In M. O. Thomas (Ed.) *TIME 2000: An international conference in mathematics education* (pp. 1–25). Auckland, New Zealand: University of Auckland.

Kaput, J., & Shaffer, D. (2002). On the development of human representational competence from an evolutionary point of view: From episodic to virtual culture. In K. Gravemeijer, R. Lehrer, B. van Oers, & L. Verschaffel (Eds.), *Symbolizing, modeling and tool use in mathematics education* (pp. 269–286). London: Kluwer.

Kaput, J., & Sims-Knight, J. E. (1983). Errors in translations to algebraic equations: Roots and implications. *Focus on Learning Problems in Mathematics, 5*(3), 63–78.

Klein, J. (1968). *Greek mathematical thought and the origins of algebra.* Cambridge, MA: MIT Press.

Kline, M. (1953). *Mathematics in western culture.* New York: Oxford University Press.

Mahoney, M. (1980). The beginnings of algebraic thought in the seventeenth century. In S. Gankroger (Ed.), *Descartes: Philosophy, mathematics and physics* (pp. 97–140, chapter 5). Sussex, England: Harvester Press.

Nemirovsky, R., Kaput, J., & Roschelle, J. (1998). *Enlarging mathematical activity from modeling phenomena to generating phenomena.* Paper presented at the 22nd Psychology of Mathematics Education Conference (Vol. 3, 287–294). Stellenbosch, South Africa.

Noss, R. (1998). New numeracies for a technological culture. *For the Learning of Mathematics, 18*(2), 2–12.

Noss, R., & Hoyles, C. (1996a). The visibility of meanings: modeling the mathematics of banking. *International Journal of Computers for Mathematical Learning,1*(17), 3–31.

Noss, R., & Hoyles, C. (1996b). *Windows on mathematical meanings: Learning cultures and computers.* Dordrecht, The Netherlands: Kluwer.

Noss, R., Pozzi, S., & Hoyles, C. (1999). Touching epistemologies: Statistics in practice. *Educational Studies in Mathematics, 40*, 25–51

Pozzi, S., Noss, R., & Hoyles, C. (1998). Tools in practice, mathematics in use. *Educational Studies in Mathematics, 36*, 105–122.

Schmandt-Besserat, D. (1978). The earliest precursor of writing. *Scientific American,* 238.

Schmandt-Besserat, D. (1992). *Before writing, from counting to cuneiform* (Vol. 1). Houston: University of Texas Press.

Shaffer, D., & Kaput, J. (1999), Mathematics and virtual culture: An evolutionary perspective on technology and mathematics education, *Educational Studies in Mathematics, 37,* 97–119.

Swetz, F. (1987). *Capitalism and arithmetic: The new math of the 15th century.* Chicago: La Salle, Open Court.

Walker, C. B. F. (1987). *Cuneiform.* London: British Museum Publications.