# 8

# More Snapshots: Kids Are Smart

As in chapter 3, I want to discuss what has already happened with our prototype computational medium, Boxer, in order to think about what may happen if we achieve a new, computationally enhanced literacy. We should be attentive to the goals and actions suggested in these snapshots, which can help speed and enhance the development of a new literacy.

In contrast to chapter 3, we now have a refined basis for considering examples. Chapters 4 and 5 developed improved ways to look at learning. Extending the discussion of cognitive and social pillars for new literacies in chapter 1, I emphasized different sorts of knowledge and competence to which we should attend. We should consider intuitive knowledge to be both a goal and a resource; we should also realize that embedding learning in extended activities that feel coherent and meaningful likewise constitutes both a critical subgoal of improving learning and also a goal in its own right. Chapters 6 and 7 focused on understanding how computational media work—how we can learn them and learn with them. This discussion shored up and extended our understanding of the material pillar of possible new literacies. Altogether, reading the messages in the examples in this chapter benefits significantly from the preparation in chapters 4–7.

The principal theme in this chapter is the cleverness we can find in people if we know where to look. In the chapter subtitle, "Kids Are Smart," I emphasize children, but of course teachers, curriculum developers, parents, people in general are included in my hopes and in the possibilities that are afforded by new computational literacies. I single out children partly because educational discussion usually (and appropriately) centers on students, but also because they are more vulnerable than others to being painted as incompetent. The new views of knowledge and the importance of activity I previously introduced are antidotes to an easy willingness on the part of adults to characterize kids as empty-headed, willing to do anything ("good" students), or unwilling to do anything ("bad" students). Whatever they are, children are not empty-headed, and it is as bad to think that some students are always unwilling to engage as it is to believe that others engage simply because they are willing.

An easy interpretation of some examples I present here is that they represent the discovery of new knowledge, intelligence, or competence that people already have, but which we have for some reason neglected to see. In a sense, this interpretation is reasonable, but a more powerful interpretation is that we are discovering possibilities for new intelligences based on a combined people-and-media thinking system. Material intelligence is not just an improved "pure" intelligence; it depends intimately on the properties of the medium that liberates it.

Unlike chapter 3, which started with an easy example, this chapter begins with a long and difficult one. Nevertheless, the example is particularly important. It introduces a new way to think about what computational media can afford, beyond what's offered by static media.

## Metarepresentation Meets a Metamedium

In 1989, during our physics course for sixth graders, we had a remarkable experience. In about five days, our sixth-grade students seemed to invent graphing as a way of depicting and thinking about motion. Graphing, as you may recall, was actually invented by Descartes, after Galileo took the first quantitative steps toward a science of motion. This was before Newton extended algebra with calculus to complete the understanding of motion that has become known as Newtonian physics. (Yes, Galileo did his work with neither algebra nor graphing! That's one reason he gets so much credit in my book.)

Can sixth-grade children invent graphing? I have been told in public at scientific meetings that children could not possibly invent graphing because such young children are too concrete or have some other limitation. I have been told that it could not possibly happen again, even if it



did happen once. The children or teacher were exceptional beyond any reasonable reality test. Nevertheless, we have the children inventing graphing on videotape; it did happen. What's more, we have repeated the activity several times since. Although it may be a bit overdramatic to say that these children "invented graphing" and hence compare them to Descartes, what happened is important to understand, and it has profound implications for the future of computational media.

The beginning of this tale takes us back to the start of our motion course. We started with a unit that taught students Boxer programming. In particular, we instructed them how to write simple programs to draw pictures on the computer screen. The first official activity about motion was an assignment to use their programming knowledge to show a few simple, real-world motions, such as what happens when you drop or toss a ball. For the dropped ball, one group developed a program that produced the picture in figure 8.1.

To most scientifically literate people, the depiction just looks wrong. It appears to show an object slowing down, but almost everybody—including this group of children—knows that objects speed up when they fall. To make matters worse, running their program reinforces the impression that something is wrong. The program takes about the same time to draw each dot and to move to the next, so as it draws the dots from top to bottom, the "ball" (graphics cursor) actually *does* slow down.

What is going on? Are these children so incompetent at programming that they can't show what they know is going on? Or have they somehow

**Figure 8.1**
A student depiction of a ball falling faster and faster.

forgotten what happens in the real world under the influence of the "artificial" computer context?

The resolution to this puzzle provided the first suggestion of the new intelligence we discovered in inventing graphing. It is surprisingly simple. When asked by a graduate student about the puzzling "slowing down" in the students' depiction, one of the pair who had worked on it said that they had not intended to "show it speeding up," but that they wanted to show "*that* it sped up." They believed they had done it quite well. More speed is shown as more dots.

If this representation is wrong in any sense, it is wrong only in using an unusual convention. Scientists usually pick equal time intervals as a basic method of cutting up events that unfold over time. From that point of view, the natural representation is to show where the ball is at the end of each time interval; dots get farther apart. You could, however, choose equal distances as a standard instead of equal times. Then you would want to show decreasing times to traverse the same distance. If we interpret the vertical axis as time, dots closer together toward the bottom could show decreasing intervals to move the constant distance.

These children were not so sophisticated as to be showing decreasing time intervals. We believe they really intended "more dots means more speed." Still, they showed a surprising sophistication. The students were not caught by the concrete and literal impulse to "show speeding up." Instead, they clearly intended to make a meaningful *representation* of speeding up. This distinction is exactly the transition from showing speeding up to showing *that* it sped up. That is, the students are not making a picture at all, but a representation in the true sense: a system of inscriptions with abstract rules of interpretation to show things about the world. Indeed, they invented a *new* representation, for they couldn't have learned this one at school!

Saying that these children designed a representation is not an example of a fancy description that applies to an everyday competence, like the bourgeois *gentilhomme* who discovered he had been speaking prose all his life. Instead, their design is a hint that children know a lot more about representations than we might suspect. *That* competence is consequential. Hold onto your hats as these students really get going!



Later in the course, when it came time to teach graphing as the most important standard representation we could plausibly teach sixth-grade students, I convinced my research group and Tina, the teacher, to try an unusual strategy. Instead of presenting graphing, why not ask the class to see what they could come up with to show motion? Tina was a good sport, although she said she expected this exercise would just show us a little about what they already knew about graphing; then we could move on and teach.

In the narrative of the students' exploring representations, I have to eliminate a lot of interesting detail. In particular, I show only what I consider landmark representations that the children developed. To illustrate their representations, I also depict only the first motion they were assigned to show—an object that gradually slows to a stop, pauses a while, then accelerates away. I also, in some cases, strip some pictographic detail from the students' representations: the students were presented with motions in the context of "cover stories." For example, the initial motion was presented as someone driving a car through the desert, stopping for a drink from a cactus beside the road, then resuming the trip. Students at first included elements of the story, for example, the cactus, which I won't reproduce.

Figure 8.2 shows an early representation, which the children called Dots. Note that, now, these students were using the standard scientific convention of equal time intervals, rather than their original "more dots means more speed." Shifting to a more conventional representation was probably due to the fact that, by this time, they had spent hours making realistic simulations, rather than representations, of motion on the computer. The dots representation can be produced from a simulation of the motion in question simply by making a dot after each step in the program.

The chalk representation in figure 8.3 highlights speed by showing it more explicitly in the length of lines rather than implicitly in the distance

**Figure 8.2**
The "dots" representation of an object slowing to a stop, pausing, then accelerating away.

**Figure 8.3**
The "chalk" representation.

**Figure 8.4**
The "sonar" representation.

**Figure 8.5**
The slant of the line represents how fast the object is moving.

between dots. This change is a suggestive step toward considering the task to be designing a representation rather than just drawing a related picture. That is, it seems to reignite the "showing *that*" idea evident in the representation of the dropping ball. Students were quite clear that measuring the length of the line was intended to be the method of determining the speed at which the object was moving at a particular time.

Figure 8.4 shows what may well be a critical step. Although one may say that Sonar, as the students called this representation, merely rotates each line of Chalk, its real innovation is to introduce a new representational resource into the pool students could use. The vertical dimension shows speed, independent of the "story line" horizontal dimension. This is really beginning to look like graphing. Still, we are not there yet.

Slants, figure 8.5, is a remarkable contribution. Jan carefully explained that the length of the lines in Slants doesn't represent anything. Instead, the slant of the line shows speed. Here is another representational resource, line "slantiness." Jan also explained that he meant the horizontal



direction to mean "as fast as it can go" and the vertical direction to represent stop. Evidently, he didn't care which way the line slanted; the same speed in the slowing-down portion of the representation (left half of the figure) was shown with slant opposite to the speeding-up segment (right half). Another student pointed out the ambiguity, and Jan acknowledged it without changing his representation.

Figure 8.6, Ts, evidently combines the ideas of Chalk and Sonar. Ts was a response to a critical difficulty with all the representations up to this point. Tina had asked several times if one could see how long the object was stopped, from which the students recognized that no representation yet could capture this piece of information. The students understood and responded to the problem. The T representation was the result. In it, the horizontal line reverts to showing speed, and the vertical line now shows time—"how long the object stays at each speed." One of the students said explicitly, "We have two dimensions; one can mean one thing and another can mean another." One of the students even went so far as to observe that you could multiply those two numbers, speed and time, to get the distance that the object moved during each depicted segment of motion. He said you could add those numbers up to find exactly how far the object had moved. I come back to this observation shortly.

Early during the third class in this sequence, Jan introduced an improvement of his slants representation, which quickly precipitated the first unambiguous graph. He described it as an "awesome idea" that combined several previous ideas and that could get "everything we want at the same time." The essence of the idea was that his slants had used only one part of the representational richness of line segments. He had used

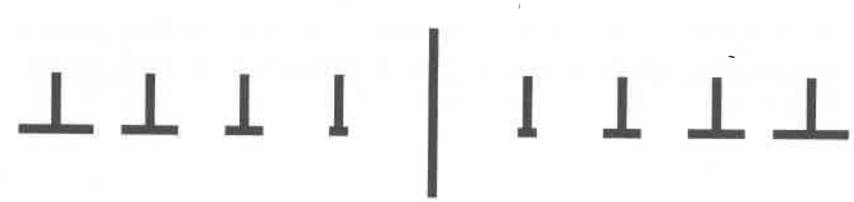

**Figure 8.6**
The "Ts" representation.

the slant, but the length was available to represent a second aspect of motion, either distance or time. At the board to explain his idea, he improvised: you could even connect all these slanted lines together! At every position along this newly unified line, you could see speed by attending to the slant of the line. Jan's first connected line had straight segments joined together, but successors showed a connected line that continuously changed slant as speed varied.

In order to understand what an excellent idea Jan had, you need to recall some aspects of Newton's great invention, calculus. One of the two central operations of the calculus, called the *derivative,* converts knowledge of the position of an object mechanically and perfectly into a knowledge of speed. If you imagine a graph of position versus time, it turns out that measuring the slant of the line provides the derivative; it converts from position to speed. So Jan has anticipated Newton in one crucial respect. He (and his fellow students) have learned the representational skill to look at a graph and see its slant at each point. Later in their schooling, those skills will become "seeing the derivative."

The other fundamental operation in calculus, the *integral* operation, is the opposite of the derivative. It converts speed into distance. The student observation—that with the T representation, one can multiply times and speeds and add them up to get total distance—is the equivalent of integration. See the discussion of calculus and, in particular, the discussion of the Fundamental Theorem of Calculus in the tick-model portion of chapter 2. Of course, it would be better if these children understood that the derivative is the opposite of the integral, and if they could use these operations independent of representational form, but what do you want from sixth-grade students *before instruction?*

Figure 8.7a comes from a standard calculus textbook. It depicts the derivative operation. Can you see Jan's slants? Figure 8.7b shows the integral. Rectangles substitute for Ts, but the process of adding up altitudes times widths is just the same.

Another student, Sean, extended Jan's continuous slants idea. Why don't you just place a grid over this line of changing slant? That way you could read numbers off it exactly. A grid would provide a two-dimensional ruler that measures the two aspects of motion (speed and time, or speed and distance) clearly and precisely (figure 8.8).



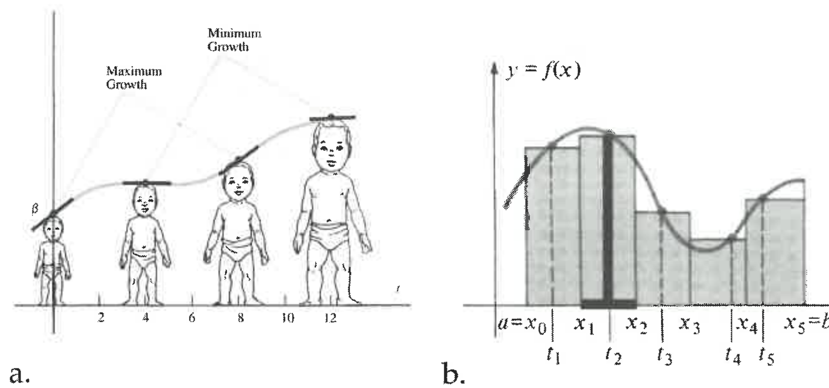


**Figure 8.7**
Illustrations from a calculus text. (*a*) Teaching students to see slant (derivative) as rate. (*b*) Integration is adding up widths times heights. A "T" is emphasized between $x_1$ and $x_2$.
*Source:* Figures from *Calculus* by Dennis Berkey, copyright © by Saunders College Publishing, reproduced by permission of the publisher.

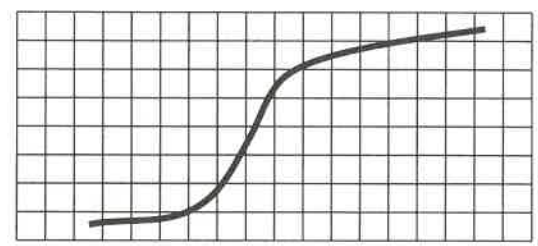

**Figure 8.8**
Laying a grid over a continuous line can allow precise reading off of two facets of the representation.

The students didn't seem to notice that this grid idea had transformed Jan's idea of showing speed via slant. They had returned to using vertical distance to show speed, but the image jointly produced by the students seemed to seize the floor, and it commanded attention in its own right. Although other representations were suggested, and although some students (gradually fewer and fewer) retained a preference for earlier representations, the students in the end voted nearly unanimously that what we would call "graphing speed versus time" was their best representation.

What is this inventing graphing activity about? Clearly, it is about bright children, but calling these students intelligent is another case of masking powerful, if unusual, knowledge. I believe they are actually more knowledgeable than abstractly bright. They know a lot intuitively about *representation,* the art of creating concrete, visual forms to express ideas. In the description above, I emphasized their inventive capabilities. They could see how lines, slants, and so on could represent things, and they could play effectively with putting multiple correspondences together. What I did not emphasize so much was their ability to critique representations. When any student proposed a new representation, others discussed both its advantages and disadvantages compared to other representations. We noticed at least a dozen different criteria for good representations, including:

- Transparency (The representation needs little explanation.)
- Compactness (All else being equal, representations that are smaller are better.)
- Precision (All else being equal, representations that allow more precise readout are better.)
- Completeness (You can get all the information you need from the representation.)
- Homogeneity (There are no extraneous symbols that don't relate to others.)
- Objectivity (It's better if making the representation can be automatic and strictly rule based, if it "could be done by a computer.")
- Faithfulness (For example, continuous representations show continuous speed changes better than discrete representations.)

The students made clear that they understood other aspects of the art of representation. They showed that they knew representations are for people. Alice, in particular, showed constant concern for whether a representation would be comprehensible to people outside this class. Jan, among others, showed particularly good ability to explain representations. He explained his slanted lines: "This [short line] has just the same slant as this [longer line], so they show the same speed. So we can use the length to show something else."

To put it in a nutshell, this activity shows that children have much more expertise with representations than most would give them credit



for. I call this knowledge *metarepresentational* because it is *about* representation. (One standard use of *meta* is "about.") To this day, I am still a bit in awe of how much metarepresentational competence these students showed. It wasn't a fluke, however. As I mentioned before, we have watched children inventing graphing several times. We haven't always been able to count on the richness and energy of our first group, but every instance of this activity has shown substantial capabilities. I don't believe there is any question that metarepresentation names a pool of competence children have, a competence that has largely been ignored.

It's nice to find some surprising understanding in children that we didn't know about before, but what does this have to do with computational media? Indeed, this activity didn't involve computers at all. (Students did want to begin building representations on the computer, but we didn't really see at the time what this activity was about. We thought it was just about graphing, and Tina kept them from exploring computational versions, at least until after the main design was finished.)

To make my points, I need to build a bridge between metarepresentational knowledge and computational media. I'll make the bridge first on the basis of principle and then with examples.

You can probably guess the rough shape of the bridge. We have identified a native pool of intuitive resources that, it seems, children develop well and fairly early—without explicit instruction. These resources are the abilities to mold and interpret visual presentations as representations for conveying information, reasoning, and all the other purposes that inscriptions, representations, and, indeed, literacies serve. Computational media can provide a context in which these competencies can grow and transform into a qualitatively different capability. In terms I introduced in chapter 1, we want to see how computational media can *implement* a different material intelligence—a new representational intelligence—starting with representational talents people seem naturally to have.

One element of a new representational intelligence seems obvious. Computers give people unprecedented control over form and space. Words and the essentially one-dimensional run of text can give way to wonderful new structures for representing. Boxer's boxes within boxes and ports to connect distant parts of a world, like hypertext, provide a hint of what is possible. Yet at least two additional substantial stages are possible with computational media. First, representations can be dynamic as well as static. All of the power evolution has conveyed to humans to interpret change in a visual presentation can be harnessed in a way that shames what can be done with text and static pictures. I'll get to a fairly elaborate new example soon, but for now we can look back to chapter 2. The boring, inert arrows and numbers of textual vectors become dynamic on the computer screen. One can watch and interpret changing vectors and write simple programs that show the meaning of a vector in controlling motion. Indeed, this humble example also illustrates the second stage of transformation that computers allow beyond textual literacy: the dimension of interaction. Vectors succeeded with our children not only because they moved and connected with motion in illuminating ways, but because they became things that children *acted on.* From the point of view of representation, computers mean that action and reaction can enter the expressive mode. You can act on text (and standard pictures) only in the limited sense of writing it down (or drawing it), and what you thus create doesn't react at all.

This part of the story seems widely recognized, at least implicitly. Unfortunately, the larger literacy implications go unappreciated. The burgeoning representational landscape is easy to see in the educational design community, where an extraordinary number of representational forms are being developed by experts. New visual representations for important ideas, from fractions to force and beyond, are an everyday occurrence. A look at professional science alerts us to the fact that this is not just a modest innovation or one only for schools. The infrastructure of science is fundamentally changing because computationally implemented representations are blossoming like flowers after a spring rain. Graphs don't just sit there anymore. They wriggle and twist with adjustable parameters. Data are not just a long list of numbers anymore. Swarms of data points fly around in changing multidimensional "slices." Visual data analysis now virtually *means* using new techniques with computers to allow people better use of their spatial/dynamic interpretive capabilities.

The less recognized part of the story is from whom, when, and how new computational representations will emerge. Let's start with "from whom." Of course experts will make new dynamic and interactive representations, but a fundamental assumption of this book is that a mass literacy surpasses a literacy of the elite. Ordinary folks—including teachers and students—should be allowed to get in on the game. Otherwise, we'll have one of those half-a-loaf consumer literacies I discussed rather than a real two-way literacy. Can we expect ordinary folks not only to control a computational medium, but to create genuinely new representations? That's where inventing graphing surprises us. Yes, they can, and probably they do all the time, without our having noticed.



"When" people can make new representations provides as much ground for surprise as "who," I predict. As I have constantly reminded you, great people and great products attract our attention far more than the commonplace, but a true computational medium can give the law of the little its revenge. I believe that creating new representational forms can become an almost everyday event, not just an eccentric, if wonderful, aberration.

There is an important technical side of making representational invention commonplace. Way back in chapter 1, I noted that some media, some literacy substrates, can comfortably enclose subforms. Text neatly encloses algebra, for example. The computer is the protean master of this trick if we design computational systems well. Not only can independent representational forms coexist in computational media, but they can interact, evolve, and change, so an essential part of what can allow representational design to emerge as an everyday activity is that computational representations can be modified or cut apart or combined. Recall the image of organic growth of software at the end of chapter 7 (see figure 7.1). A producer computational medium is a *metamedium,* in which hundreds of microrepresentational forms can be created, combined, and extended constantly. Not every child will create a completely new representation everyday, but very often everyone can make easy extensions, combinations, and modifications; occasionally, really new things will appear. Representation will be a richly tooled, flexible, adaptive, and improvisatory activity far beyond what exists now with textual literacy.

I want to be clear on preconditions for this transformation to happen. First, we can't use just any old computer system to realize it. A consumer-only medium, for example, won't support it. Second, the range of the representational forms that are allowed will be highly constrained by the properties of the medium. If the world standardizes what is now called *multimedia* as its new literacy base, I fear new representational forms will be limited to successor variations of rock video. Naturally, I believe that vectors and hundreds of other such forms can enhance learning science and mathematics, but many current versions of computational media won't support analytic, science-relevant representation.

Finally, the emergence of representation as an explicit, valued activity—as any important component of literacy—will be a cultural accomplishment. A "literature" of representational forms must emerge, as well as an awareness of and commitment to their value. It is a plain and simple fact that technology by itself isn't enough.

I used the example of inventing graphing to argue that metarepresentation is a fundamental competence humans have. Then I argued abstractly that computational media can liberate and extend this intelligence in particular ways. Now I want to give some specific examples to put meat on these wonderful but bare bones.

Let me start with examples from some very new work. We have recently started looking specifically at student representational creativity in computational media. In particular, we taught a class of middle and high school students mainly about scientific visualization—making pictures of scientific data for visual analysis. Visualization is a part of professional scientific practice that has been radically changed by computational representations. One fairly simple but important technique is colorizing images to bring out particular detail or relations.

Instead of giving students closed scientific tools, we built some open tools in Boxer so that they could get their hands on the guts of the programs. In particular, we made it easy for them to design and construct the palettes of colors used to display images. Figure 8.9 shows an image taken from the Hubble telescope (top left), modified by one student with different palettes. The first thing Sam did was to change the color at the top of the scale, the color that showed the brightest parts of the image. In these pictures, all data that are above a certain brightness show with the color at the top of the scale. Any variation in brightness above this level just can't show in the representation; there are no more colors. In effect, Sam is showing what part of the picture has been completely washed out by the limits of the display system. He picked red for that



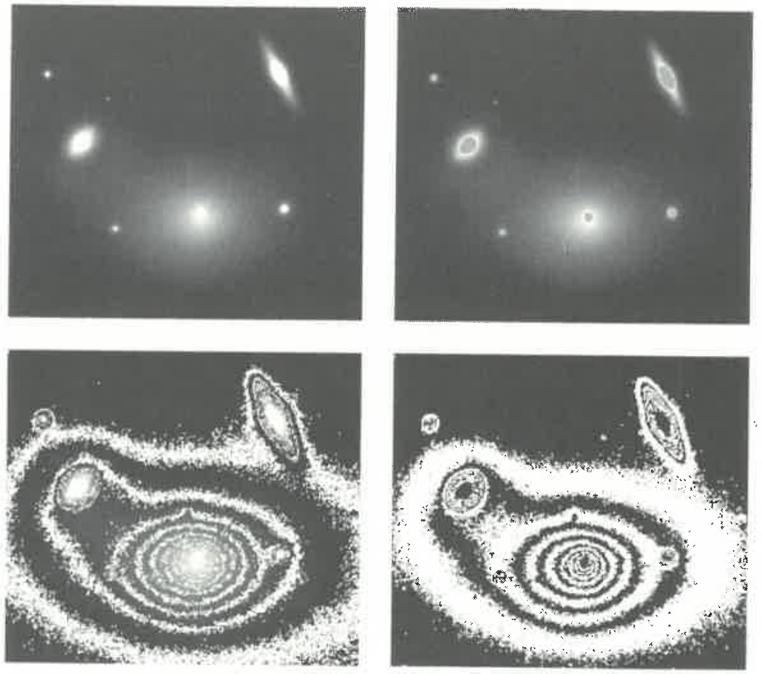

**Figure 8.9**
Images from the Hubble satellite telescope, modified by a student.

color, and you have to imagine the small, gray regions in the center of the bright spots (top-right image) standing out vividly. I have not seen the same idea in any professional system. Perhaps experts don't need a warning, but Sam's idea dramatically shows which parts of the picture have no detail.

The lower-left panel of figure 8.9 shows another of Sam's innovations. Again, we are limited to black-and-white printing in this book, so your imagination is important. In order to show shape and detail better, Sam built a palette that alternates a sequence of colors with a constant background color. (Successively brighter parts of the image "climb up" Sam's sequence of colors, touching base with a constant background color in between.) The result is something like a topographical map, in which contours show places of constant altitude. Here, the contours show constant brightness. The shapes of objects are much easier to see in this representation, as is how quickly light is getting brighter. (Rings that are closer together show that the image is getting brighter more quickly.) Sam's idea is better than standard contour maps in at least one respect. The color coding he put between contour lines (which were made of the constant "background color") shows brightness unambiguously. In contrast, if you use only black-and-white contour lines, you can't tell whether brightness is growing or waning; either direction results in more rings. The black-and-white contours of the last frame (lower right) can't distinguish between a "donut" nebula—which gets brighter, then darker toward its center—from a circular object that gets continuously brighter. The light gray tones near the center of the biggest galaxy in Sam's contour image (the lower-left image) make it unambiguously clear that the center of the galaxy is brighter than the surround.

What you can't see at all in these images is the spectacular impression made by the colors Sam chose. Sam was an artist, and he spent almost as much effort making things pretty as making them scientifically better. Think "committed learning," and you'll understand why his pursuing art as well as science didn't bother us at all.

One of Sam's colleagues, Mohammed, made use of the palette system for an even more personal purpose. He modified a computer game he had written so that the numerical score was not only shown but color coded. That way the game player could tell how far she had progressed without taking her eyes off the action. Besides being a nice representational idea, Mohammed's color coding the game's score both suggests the power of personalization and reminds us of the law of the little. Mohammed's creation may not be grand, but it is his. More, it suggests the thousands of little innovations awaiting students' creativity when any representational fragment that already exists in the medium can become part of their private constructions.

The final example of students' invented representations in computational media is not visually spectacular, but it is an intellectually spectacular and huge project by a high school student.

Ted got his first experience with Boxer in Henri's Boxer statistics course, which I mentioned in chapter 3. Ted, like Mickey, the young Boxer manual writer, was dedicated to helping others, and he continued working with Boxer after the course's end. He wanted to design and build



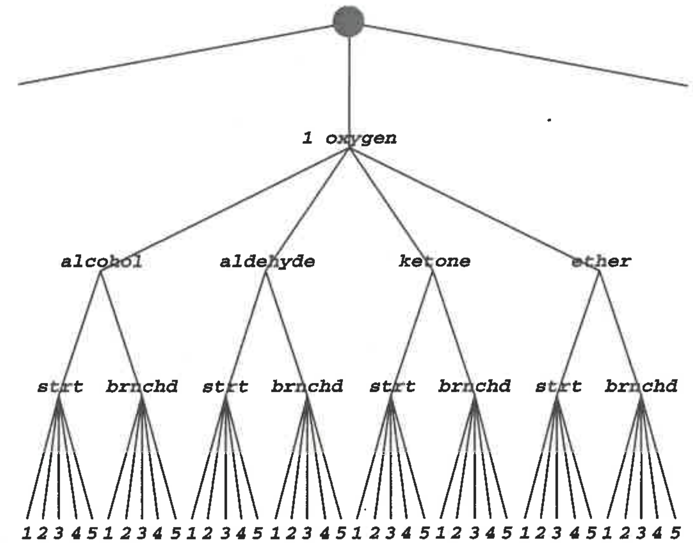


**Figure 8.10**
Part of a dynamic figure that illustrates an algorithm for naming molecules.

tools to help students and teachers approach particular subjects. One program he wrote, of which I'll show a tiny part, was called the *molecular toolkit,* and it provided resources for students and teachers to make interactive presentations about chemistry. Parts of the program could show pictures of molecules based on their formulas, compute such things as the weight of a molecule, and even name molecules by their formulas.

Figure 8.10 shows one-third of a representation Ted invented to show how the naming process worked. The representation is a decision tree that you work down, one branch at a time, by answering certain questions. At the top level, you need to count the number of oxygen atoms in the molecule. The portion of the tree in figure 8.10 corresponds to one oxygen molecule; zero and two oxygens are branches not shown to the left and right. At the second level, you have to decide whether the molecule is an alcohol, an aldehyde, a ketone, or an ether, and so on at lower levels. While Ted's program finds the name of a particular molecule, the ball at the top of figure 8.10 moves down the various arms of the tree, and text explains what is going on. When the ball gets to the bottom, the program spits out the scientific name of the molecule: Methane ($CH_4$), ethane ($C_2H_6$), methanol ($CH_3OH$), ethanol ($C_2H_5OH$), ethene ($C_2H_4$), 2-propanone ($C_3H_6O$, with the oxygen double bonded to the middle carbon), methylpropene. . . .

This is a wonderful dynamic representation of naming molecules. I have never seen any version of it (even a static one) in a textbook, and it almost certainly was Ted's invention. Even better, Ted clearly understood that his overall project was providing tools that others would extend and modify. As I mentioned, he explicitly considered this a toolkit to build presentations, not a "teaching program" on its own. I believe he intuitively understood the flexibility and organic growth of representations and microrepresentations in a computational medium about which I spoke earlier.

Ted's program is among the best-organized Boxer programs I have seen. His data structures are elegant. For example, he invented a new notation for organic molecules that is easy for humans both to understand and to produce, as well as computationally clean and efficient.

Figure 8.11 shows an example of Ted's Boxer code corresponding to the decision branch based on number of oxygens. The command **ifs**

```
ifs
  oxygens = 0    trace-arm 1 1
                 inform
                   The molecule has no oxygen atoms.
                 hydrocarbons
  oxygens = 1    trace-arm 1 2
                 inform
                   The molecule has one oxygen atom.
                 alc-ald-ket-eth
  oxygens = 2    trace-arm 1 3
                 inform
                   The molecule has two oxygen atoms.
                 carboxylic-acids-esters
```

**Figure 8.11**
A part of Ted's program for naming molecules, displaying the algorithm for naming.



presents a number of "if" possibilities (**if oxygens = 0**, etc.) and what to do if those possibilities are realized. In each action branch, you can see the command to animate the decision tree by moving the ball down the appropriate branch (**trace-arm**), the textual information presented to the viewer (**inform**), and the next decision (e.g., **alc-ald-ket-eth** makes the decision about the second level of figure 8.10). A well-structured program is a representational achievement in its own right. Not only does such a program make it easy for others to inspect, understand, and modify the program itself, but it can even be a directly instructional representation. I am not sure that Ted's program is the best representation of the molecule-naming algorithm, but it is pretty good. It contains all the detail missing from the visual tree. On the other hand, the programs for the fractal shapes designed by the students in Henri's infinity class almost certainly *are* the best representations I know for conceiving and knowing how to build those shapes—provided, of course, you are programming literate.

The argument I am making has several parts. By this time, I hope you'll concede the first point—that the computer is the protean mother of metarepresentational systems. It surpasses text and inert, noninteractive graphics as if they were baby steps and not the giant leaps of human material intelligence we know they have been.

The next point is that representations are a fundamental part of science. This part is still not at all the most difficult, although it is not a common story. Galileo invented pictorial, quantitative ways of thinking about motion that supported his investigations. Descartes invented analytic geometry and graphing. Newton and Leibniz invented the calculus and, simultaneously, ways of denoting their new ideas, ways that are conceptually suggestive and easy to manipulate and reason with. Richard Feynman invented Feynman diagrams to denote interactions of elementary particles. Lest the grand obscure the everyday, I note (again) that even grand representational accomplishments really happened gradually, in little bits and pieces, over years of use by the scientific community. Furthermore, scientists are constantly adapting and inventing, in dribs and drabs, ways of making ideas clear to others and ways to help themselves think.

Next generation students will have a bewildering array of new representational forms to master if they are to learn the science scientists are now beginning to practice. Luckily, from inventing graphing and similar experiences, we can have more confidence that students are up to the job, but here's the really difficult part of the argument that I am making: I believe we need to see putting computers and students together as liberating students' own creativity at designing representational forms, just as it has done for scientists.

The role of representational creativity in learning is just now beginning to be demonstrated scientifically. Examples can't prove the case, but look at Ted's remarkable accomplishment. He invented two marvelous new representations of the process of naming molecules. His dynamic decision tree is impressive enough, but the program that actually carries out the naming is a showstopper. Could we imagine his doing such a thing without understanding every nook and cranny of the process of naming molecules? Technical reexpression of ideas, like what Ted did, tests and improves understanding at each step in the process of construction. Success indicates a mastery that surpasses anything an ordinary school test could hope to show. (Also—although I'm getting ahead of myself—constructing representations like this is a personal accomplishment that puts typical school tests to shame in the personal value it can have for a student.)

Ted's case is too extraordinary to be convincing about the everyday future. That's the trouble with a short text in which an author is limited to a few examples—they had better be dramatic (hence, unconvincing as examples of things that will become ordinary). Sam and Mohammed's little representational achievements help, but they may also be unconvincing: it is relatively easy to imagine students doing those things, but is profound new learning at issue?

Let me sketch a pattern that may be easier to see than these examples as a new and important wrinkle in science education. Essays have been a staple of literary instruction for as long as there has been organized instruction. What happens when a student writes an essay? She needs to think hard about the subject, formulate a personal position, and systematically lay out an argument for that position. As an extra bonus, the teacher gets a great deal of insight into how the student is thinking, which would be impossible in a short test or conversation.

Essays are not nearly so popular in science or mathematics, probably because natural language is not sufficiently tuned toward the particular



nuances of meaning that are important for science. Equations? They are better in some ways, perhaps, but hopelessly sparse and too impersonal to be really expressive of a student's way of understanding scientific ideas. But what about a *computational essay?* A computational essay can use text to describe and hypertext to organize. It can have diagrams, even moving diagrams, like Ted's decision tree. It can have little programs for the reader to play with, like the ones the students in Henri's infinity class created. Indeed, their final projects were nice examples of a budding genre of computational essay. A computational essay can have dynamic models. It can also include any sort of tool or computational subsystem the teacher or curriculum developer might supply to help students think and work in a particular area, modified appropriately by the student for her expressive purposes. A nice subgenre is the computational essay expressing the results of a scientific investigation, which is particularly useful to hand to other students so that they can examine the data that might be enclosed, using their own tools or models. Other students can also "borrow" tools or microrepresentations invented by their colleagues. Another subgenre already in use, at least among some researchers, is a computational essay that a student writes in order to teach another student about some subject matter. Above all else, a computational essay invites and empowers students to innovate representationally to help themselves think and to express their ideas about scientific and mathematical subjects.

We have had limited experience with computational essays as a way of learning. Sadly, exploring all the promising possibilities of computational media at once is just not feasible, especially with limited resources, but what has become fairly systematic in our work is to make each exploratory microworld also a place where students can easily collect and annotate their work to explain it to others.

One of the most exciting types of computational essays, which I call a *knowledge space,* we have only barely begun to investigate. A knowledge space explores the organization and relationships of the many parts of understanding any particular scientific topic. It's best to think of constructing a knowledge space as a culminating activity in which students or a whole class working together use a spatial metaphor to organize and show relationships they feel are critical. Of course, if they are really learning, students may already have developed intuitive mental maps of the connections between the various aspects of what they learn about photosynthesis or organic molecules or Newton's laws, but identifying the core ideas, finding an explicit and systematic external form in which to relate them to each other, linking in more peripheral ideas—pushing toward a completeness in showing and connecting things, which text or intuitive thinking cannot reach—are the distinctive activities of making a knowledge space.

## An Activities Perspective on Metarepresentation

I organized my description of the inventing graphing activity as a sequence of representational ideas, which is a beautiful way to tell the story because it highlights the emergence of recognizably powerful, important knowledge out of previously unrecognized human competence. That is a central theme of this chapter. There are, however, always other ways to tell a story. In this case, my research group did not even know there would be a good knowledge story in inventing graphing when we started. It was just, after all, a curricular lark. Instead, we stood at attention and turned on the video cameras for an entirely different reason.

In simplest terms, what attracted us first to collect video data about this event was the remarkable enthusiasm and interest it engendered in the students. From the beginning, we could see that something special was going on. To put it in a different way, we could smell a good example of committed learning coming. In a larger framework, the ideas story of inventing graphing misses the critical activities perspective that we developed in chapters 4 and 5 to complement a knowledge perspective on learning. I want to redress this omission, at least briefly. I'll use inventing graphing exclusively, even though, as I hinted, the other student representational examples have interesting activity stories about them as well.

Let me start with a few vignettes and descriptions to give a sense of how we perceived the inventing graphing episode and, more importantly, how it must have felt for the participants.

### Enthusiasm

The enthusiasm the students brought to this activity was extraordinary, almost from beginning to end. They were champing at the bit to get in and explain their ideas. Commitment and passion ran high. The classroom was often bustling to the point of chaos. Many times Tina couldn't even get into the conversation, and she would repeatedly try, wait graciously if rebuffed until the students quieted down, then step in to make her points and to help direct the inquiry. One of Tina's special strengths was cultivating the personal investment of students, and she systematically tried to create situations in which she was irrelevant to the activity. At this stage of scientific study, I think clever teachers like Tina know much more about activities and how to cultivate group committed learning than researchers do.



### Participation

This was not a class of super students, even if they were bright. They all had different inclinations and styles. At least two of the students were very quiet, and a couple seemed to want to dominate every activity. One unusual thing about inventing graphing was that it elicited much more even participation than other activities. Even the quietest student made critical contributions, and at various times all wanted to have the attention of the class. One of the quieter girls in the class took on the responsibility of sticking up for the poor "average person" who might walk into the classroom and find a jumble of uninterpretable squiggles on the board. How would a visitor, she kept asking, make any sense of these representations? She had the passion of a political activist defending the rights of the downtrodden.

### Ownership

The "dance of ownership," as we came to call it, was amazing during inventing graphing. Students were proud of their accomplishments. Sue added a copyright notice to many of her drawings. I recall her walking past the camera at the end of one class and deliberately asking the video viewer, "Isn't my drawing nice?"

Ownership was also communal, however. Students borrowed others' ideas freely, usually with acknowledgment. Here's where the dance metaphor comes in—the ebb and flow of students' leading, following, building on each other's contributions. They could see something developing that was too big for any one of them to take credit for alone. The teacher was dancing, too. In pointing out Tina's attention to student ownership, I don't want to imply she did not want to be part of the conversation. She had ideas she wanted to share, her own ideas about "good representations," and several times I was stunned at how direct and imperious her criticisms of student ideas were, but students took the criticism with remarkable confidence. They seemed to acknowledge good points when Tina made them, and they certainly knew she could make them stick if she wanted to, but they often conveyed the sense that they were not convinced and that their conviction mattered. In retrospect, I feel Tina's skill was not either in being direct or in withdrawing, but in knowing when to act in which of these ways.

**Taking It Home**

We saw many signs that this activity carried on outside of class. The most vivid for me was Jan's "awesome idea," adapting slants to do "everything at once." His awesome idea came right at the beginning of class. He had to have come to class with the idea because it didn't follow any context set by the discussion to that point. Classroom barrier breaking is a fingerprint of committed learning.

**Momentum**

Probably the most fundamental question about any activity is whether it can continue on its own energy. Inventing graphing had plenty of such energy, but any extended activity is patterned with waves of greater and lesser intensity. The teacher's role is particularly important in the waning periods. We could almost see Tina selecting and introducing a new hill to aim toward for the class's activity roller coaster, a hill that would suit their current skills and current attention, but also one that would stress their current understanding and thus develop the ability to climb new representational hills: "Can you do this with your representation?" "Would you want to do that?" "Let's just spend some time practicing this or that representation."

An emblem of the success of this entire activity came when Tina decided it was time to quit and go on to other things. She summarized what they had accomplished and why it was important, and then she began to explain what the class would be doing next. A student raised his hand as if to ask for clarification, and Tina turned to him, but he started again asking about how they could pursue the last issue that had come up in



inventing graphing! The whole class laughed at the non sequitur and the fact that Tina evidently was not succeeding in shutting down the discussion. As the bell rang, the class dispersed, and as they left, a small group discussed among themselves whether they could get a good answer to their question from NASA scientists.

**Childlike**

It is critically important that a good activity be continuous with the lives of the participants. Knowledge is important, of course, and in this case, inventing graphing couldn't have happened without the untutored metarepresentational expertise of these students, but, just as important, this activity picked up and extended patterns of engagement. One fundamental thread was how the students attended sincerely to the contributions of their peers. School is ordinarily not like this. "Right and wrong" is always the point, and the teacher is the knighted arbiter. The teacher is even obliged to take up this role, or students will complain she is not teaching. Tina worked very hard in making students' ideas the focus for discussion. She talked to us about how difficult it was to turn students' attention away from the teacher and away from "right answers," especially in a school where teacher-centered instruction was the norm, but she succeeded in making student ideas the center of the discussion with this set of children, admitting occasional lapses.

I want to emphasize continuity by explicitly acknowledging the components of inventing graphing that were not adultlike and scientific. The participants in the activity were sixth-grade students, after all. There was plenty of teasing, complaining, and general off-task dallying. As a miniature example, when some enthusiastic exchanges broke out, one student egged his peers on with, "Let's yell at each other and stuff!" Stories about children and learning that paint them purely as little scientists leave me cold. Those stories are at best well-intentioned lies or proof that, whatever is happening, committed learning doesn't live at that address.

Final point: Inventing graphing illustrates one of the general ideas I introduced in chapter 5. An enrichment frame turns knowledge into an activity structure. Inventing graphing illustrates one of the simplest and most often successful enrichment frames: instead of showing students something, have them design it. They will surely learn more about how the thing they design serves its purpose—the function of the intended structure—and they will also learn a lot about alternatives that may be desirable in particular situations.

This painfully simple idea, "have them design it," has been a constant theme in our work. Student invention works far better with a computational medium and in tool-rich culture than with other media in other contexts. I'm sorely tempted to tell you another story about how a group of our high school students designed Newton's laws! Couldn't happen? Don't be so sure.

Did these sixth-grade students learn more about graphing than other students who might have gotten a four-day lecture and seat exercises? Perhaps. Perhaps not. But they certainly explored the space of representational possibilities more thoroughly. More directly to the activity point, they had a rollicking good time being creative around important scientific ideas.

## Dynamic Representations: Intuitive Knowledge in Action

Metarepresentational competence is a lovely example of important intuitive knowledge on which computational media can build, but some readers may find it a bit too rich to wrap their heads around—like having both apple pie and cheesecake for dessert. The following example is not nearly so outer space exotic, but it makes a similar point about the adequacy of a medium for engaging natural human competence and about implementing new intelligence.

This example concerns the capacity people have for giving meaning to dynamic presentations. This is only a tiny part of what computational media can add to textual media, yet it happens to be particularly important because of how good people are at understanding moving things.

Imagine we have control of a magical railroad flatcar. It isn't *so* magical; it's just that, instead of being stuck on a one-dimensional track, it can move around in any way that we command it (except up in the air—it can't levitate). Imagine also that there is a little robot on top of the flatcar. You might think about this robot as a programmable "turtle" that you can command to move around in whatever way you want. (No



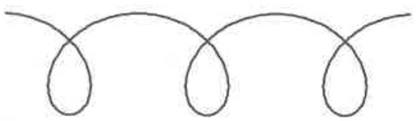

**Figure 8.12**
The path of a turtle moving in a circle but also being moved along in a straight line.

levitating turtles, either.) Suppose we ask the turtle to move continuously around in a circle on top of the flatcar. That's simple enough. But what if the flatcar is moving too? To start simply, let's say the flatcar is moving smoothly along in a straight line. Let's say it's going to Chicago. Now, if you were to watch the turtle from an unmoving perch, above, it would still be moving in circles, but also going along with the flatcar to Chicago at the same time. Figure 8.12 shows what you would see for the path of the turtle.

Physicists and mathematicians would call this a problem of composing motions. *Composing,* in this case, simply means putting together. The loopy path in figure 8.12 is the result of composing a circular motion with a straight line motion. Composing motions is usually introduced in high school or early in college after all sorts of mathematical preparation. I won't tire you with the details, but understanding the topic mathematically entails a number of things—for example, that the velocity of the turtle at any particular moment is the vector sum of the velocity of the flatcar plus the velocity of the turtle in its circular motion, ignoring the flatcar. How far can children come in understanding composing motions without the usual formal preparation?

I will tell this story in miniature by showing what one pair of sixth-grade students did with one fairly advanced problem. This pair of students was in a laboratory study before our full-fledged course. Let me show you just a little of what the students did before they got to the problem I want to present in some detail.

Figure 8.13a schematically represents the case where the flatcar is moving from left to right, and the turtle is also moving from left to right on top of the flatcar, at the same speed. Intuitively, this is a very simple case. Any number of p-prims can get the right prediction for the motion of the

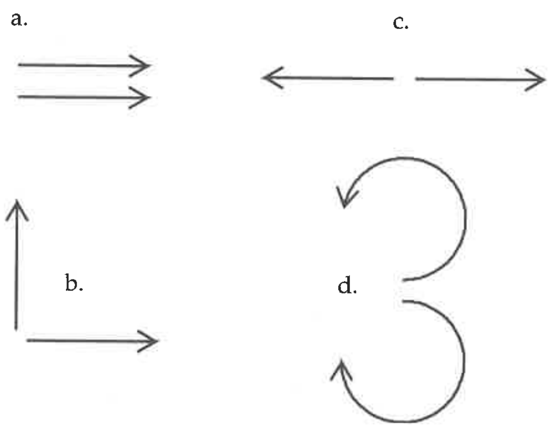


**Figure 8.13**
Four pairs of motions.

turtle as viewed from a fixed viewpoint. For example, you can see both motions as influences on the turtle's global motion and, in this case, they clearly "reinforce" each other. The net result should therefore be a "stronger" motion of the same sort; that is, the result should be faster motion to the right. A mathematical description of this case is colinear (in the same direction) vector addition. Children almost always guess the right magnitude, that the turtle's net motion is twice its own or the flatcar's motion.

Figure 8.13b represents a case that is still fairly easy. Here, the flatcar is moving toward the right, but the turtle is crawling in a perpendicular direction, across the flatcar (up, in figure 8.13b). Surprisingly, most children in upper elementary school also have a good sense of this case. They see the need for some kind of "compromise." The obvious compromise is a direction at 45 degrees, between the motions of the turtle itself and the flatcar. Even better, children can almost always extend this insight to the case where either the turtle or the flatcar is moving faster than the other. They guess correctly a compromise, but one oriented more toward the stronger motion. The cases of figure 8.13b with equal and unequal motion are excellent qualitative versions of vector addition in the "orthogonal" (perpendicular) case. When children do have trouble with perpendicular motions, they usually guess, for example, that the net motion



will alternate small motions to the right with small motions in the perpendicular direction. Actually, alternating small motions in one direction and another is a beautiful way to think about composing motion in general, but you need to think about these little motions as being really tiny. Calculus allows you to think of them as being infinitesimally small.

The last preliminary case is where the motions are opposite, as depicted in figure 8.13c. Here the correct p-prim is *canceling*. Canceling happens to be a very compelling idea, and it usually takes little or no time for students to guess that there may be canceling in this case. Students always see canceling in retrospect, if they don't guess it. Concretizing canceling with a little mental model is helpful. Imagine walking down the up escalator at exactly the same rate as the escalator is moving up. The escalator takes you up one step length, and you take a step down. The net result is that you just don't move. Children can usually generalize this situation to the case where one motion is faster than the other: "The stronger wins."

Carol and Ming had worked through the motions described above and a few others besides using a little microworld that I had written in Boxer. The microworld allowed students to look at each of the motions individually, then I asked them to guess what would happen when we had both motions at once. Taking time to think through why whatever happens actually does happen is important, especially in cases where it is unexpected, but taking time to consider "why" is important even in the case where students first guess correctly. The microworld also has facilities to help students reflect. For example, they can slow the motions down or run them one little step at a time, or they can show the paths of each part of the motion and the path of the net motion. Seeing the path of the turtle being generated, as if it were dragging a paintbrush along as it moves on the flatcar, is especially illuminating. For the perpendicular motions, the turtle's path will be an increasingly long vertical line ("vertical" means across the flatcar) that moves with the flatcar to the right. Don't I wish you were reading this book in a computational medium so you could see the simplifying visual effect!

Now, here is the piece de resistance of child expertise. I call the motion *movie reels*, as depicted in figure 8.13d. The flatcar is moving in a circle in a clockwise direction, starting its motion toward the right. The turtle is moving similarly, but in a counterclockwise direction. Carol and Ming looked at the individual motions. Ming provided the first solution move. Wouldn't the motions cancel? After all, they are opposite, just like moving in opposite straight-line motions. Ming was casually moving his hands around in circles side by side in opposite directions.

Carol started to agree, but then she began looking at her index fingers, which she was moving more and more carefully out in front of her, each simulating one of the two movie reel motions. She noticed that the two motions start out in the same direction. "Hold it!" She described how the flatcar and robot both start moving toward the right. Then she gestured and continued, "So it's actually going to make a . . . 'cause . . . It's going to make a line!" Carol is correct. The two circles combine into a motion where the turtle simply moves back and forth in a horizontal line.

Carol made several superb moves all at the same time in leaping to the line solution. First, she followed Ming's move of considering the motions as going together abstractly, rather than trying literally to simulate the flatcar carrying the turtle. That is, she just looked at the two motions and tried to combine them. That you can do this is not at all obvious. The flatcar is carrying the turtle, not the other way around. Carol was looking at this situation as if it didn't matter who was carrying whom. Actually, this is a theorem. You can reverse the motions of the flatcar and the turtle, and you get exactly the same net result. Ming and Carol seemed to have guessed this theorem, possibly based on experience, or maybe they accidentally fell into using it implicitly. If I had asked them, though, I'm sure they could not have articulated and certainly could not have proved this theorem. In any case, Ming and Carol had a wonderful simplification that will never fail, and it will help them solve many complicated motion composition problems.

Carol did something else clever. She started looking at motion locally—that is, in short intervals of time—rather than the global consideration of opposite circles as canceling, which Ming demonstrated in his casual gesturing of opposite circles. Carol started focusing on the velocity at a particular time, at the beginning of the circular motions. She saw one of the main points of calculus, as I described in chapter 1—that whenever motion isn't uniform, you always need to say *when* in order to say anything about speed or direction. Looking at the beginning of the movie reels motion, she saw a case just like the one given in figure 8.13a, both motions going the same direction.



The final stroke of genius in Carol's analysis is that she saw canceling *in addition to* reinforcing. As became clear in asking her to explain her reasoning to Ming, Carol saw that, in the vertical dimension (again, vertical as the motions are portrayed on this page and on the computer screen), the reverse circle motions are actually mirror images. The turtle moves from the center to the top, while the flatcar moves from the center to the bottom, and so on in synchrony. Thus, in net, the circles add their motions in the horizontal direction, and their motions cancel in the vertical direction, leaving a simple back-and-forth motion in a straight line. Note that canceling does fit in this situation after all, although subtly.

What are the lessons of this example? First, some nonlessons. Ming and Carol are not "the average case." The movie reels problem is quite difficult, and not all elementary school students can master it, at least not as quickly and elegantly as these two, so I do not mean to imply that all learning can be as quick and faultless as in this example. On the other hand—and this *is* a lesson—all students with whom we have come in contact have strong intuitive resources of the same sort as Ming and Carol. That is, they can reason well about motions. Competence with motions is actually not surprising at all if you think that humans must have evolved with strong dynamic spatial visualization skills in order throw rocks and spears effectively, to block or catch moving objects, to navigate while running, and so on. Every student who used this microworld could master all of the ways of thinking about motion I talked about in preparation for the moving reels motion: canceling, reinforcing, compromise. They could even think about motions separately in vertical and horizontal components, and they could pick instants to focus on rather than using a global gestalt of motion. All of this is within the reach of essentially every late elementary school student.

On the basis of strong intuitive dynamic visual reasoning, we see where computational media can pick up from text and other static media to take us forward. This little microworld, which I programmed in about a day, turns out to be plenty of help for students to learn quite a lot. There is no magic. Just find a good source of (intuitive) knowledge, provide it some computationally enhanced experience, a few ideas and time to reflect, and quite a lot of learning happens. When it comes to learning about motion, a fine regime of competence exists, and teaching may capitalize on it.

Allow me to lay out some absolutely typical patterns of learning on the basis of intuitive knowledge. First, students don't always guess correctly. Intuitive knowledge is not reliable in the way that more systematic knowledge can be, but, critically, there is almost always a more adequate intuitive conceptualization available if the first guess fails. Students who don't see canceling or compromise at first come quickly to feel these ways of thinking are sensible after seeing what actually happens and reflecting on it. Seeing sense after the fact is just like what happened with my intuitive electronics knowledge when I got into physics class. I didn't know what was taught there, but I could think about it and integrate my intuitive knowledge appropriately.

Note how easily one could paint a negative picture of children. Without applying a little care and patience, without looking to see what resources students have available, and without giving them a situation that evokes those resources, you could find that young students don't know much about composing motions. In particular, I can guarantee that many students will think movie reels cancel the first time they see them, but you now know what we learned from students such as Carol and Ming: canceling just needs a little encouragement and help fitting into this situation. Then canceling becomes not a mistake, but part of a correct and powerful conceptualization. I don't think I can overemphasize that intuitive knowledge is not perfect, but it provides resources that designers and teachers need to know about and use.

A more subtle pattern of use of intuitive knowledge is in Ming and Carol's implicit use of the theorem that the motions of the turtle and flatcar are interchangeable with the same result. This guess happens to be brilliant because it works. On other occasions, kids will make brilliant guesses that don't work. The trick for us as educators is to collect those brilliant guesses—right or wrong—and find how we can use them productively. Even in the case where the brilliant guess has been elicited in an environment where it works, there is more learning to do. Ming and Carol couldn't articulate or justify the interchangeability theorem, so even they have an important opportunity to extend and refine what they stumbled across. The first step is to realize that the theorem might not be true. Knowledge often arises from the recognition of ignorance. This is a nice example where the more you "know" (like Carol and Ming apparently did in guessing the interchangeability theorem), the more you are in a position to learn. Recall what learning in the regime of competence means.



Another typical pattern of intuitive learning is that children initially conceptualize reinforcing, compromise, and canceling separately. They're just different phenomena. From a mathematical point of view, they are all examples of vector addition. Thus, eventually students will unify many disparate understandings under the banner of a single scientific idea. Recall the way algebra unified Galileo's six fundamental theorems about motion. I didn't show Ming and Carol getting to that level of understanding. Indeed, they did not get there; they were just subjects in a laboratory study, not students in our motion class, so it would constitute a goal for their extended learning about motion. Nonetheless, they had to learn a lot even to get to the stage they reached. Intuitive knowledge is more the basis for advancement than it is a given fact of life. Intuitive knowledge is dynamic and generative. Looking forward, what Ming and Carol did learn here will not be replaced or redone by "formal understanding." Instead, what they learned will make "formal learning" seem sensible and easy.

A final episode with Carol and Ming serves to emphasize the critical link between this particular child expertise and computational media. One of the puzzles I put to them was where the turtle goes uniformly in a straight line, but the flatcar "falls" downward. From the tick model in chapter 2, you might recall, falling is just going downward a little more each tick. (In other words, speed just increases in the simplest possible way as time marches on.) These students had an excellent science teacher at school. He had taught them about falling. When one of them asked me whether the "falling" of the flatcar was actually like gravity, like real falling, I put the question back to them. After a moment's thought, they said no. Their teacher had taught them that falling "goes like squares." (I am pretty sure what he taught them is that falling is measured by acceleration, thirty-two feet per second *squared*.) Carol was pretty sure (and correct) that the falling she saw on the screen was "just going one more each time," rather than 1, 4, 9, 16, . . . . How poignant that an excellent teacher with wonderful students but with the wrong medium had failed to convey a powerful, simple, and intuitively apprehensible (with the right medium!) idea. Falling is precisely "going one more each time," the pattern Carol saw easily in the microworld's falling motion. Their teacher probably never showed them a fall in slow motion, certainly never with replay and analytic facilities built into their experience, as with the composing motions microworld, and absolutely never showed them the tick model.

### Prodigious Products

This final example is meant to complement the two given above. Instead of knowledge, I emphasize here some things that relate to technical aspects of a medium. In particular, I bring back issues concerning making things with computational media. These issues are in the family of modifiability, extendibility, adaptability, organic growth, cumulativity, long lines of evolutionary development for software, alternate production niches for software, communities of tool builders and sharers, and so on. In addition, I discuss some activity-relevant issues.

The main focus of attention is the product of two energetic young men in our sixth-grade motion course. This product was part of an independent project on which they worked toward the end of the course. They had many ideas and much competence on which to draw. Nonetheless, the project struck me as remarkable.

Sean and Bob had produced a huge system—a graphing adventure game. The story line, which the students had written into the introduction, starts with the fact that you are a friend of a movie star named Michael. Michael is doing commercials for the 9.3167289 Lives cat food company, and someone has written a "swear word" into the cue cards. Now, Michael is running for his life from the evil corporate bosses, and he asks you to help him. The game progresses through a long series of adventures, where at each stage a motion is described that can extricate Michael from a particular fix. You have to choose a graph that corresponded to the life-saving motion. If you do not select correctly, you and Michael die a horrible death, and the sequence of adventures starts all over again.

Here's a sample problematic situation—in fact, the very first challenge in the game. You are driving with Michael out of the parking lot and



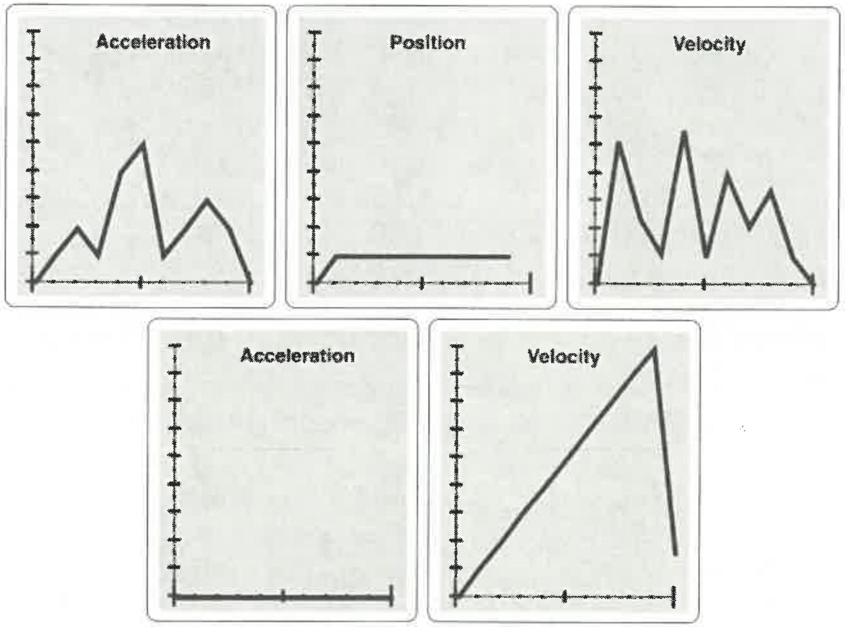


**Figure 8.14**
Graphs from the graphing adventure game.

need to speed past the guard post. Figure 8.14 shows your choices. The bottom row, left, is correct. It shows 0 acceleration, which is equivalent to moving at constant speed. If you choose top left, you get:

Are you sure you didn't
drink too much coffee?
Your foot is tap dancing
on the pedal! You swerve
into the guard post, and . . .

*long pause*

KABOOM!!

Notice that this text gives the player informative feedback about the selected graph. In fact, making this game educational, thus acceptable as a class project, involved a long series of negotiations with their teacher. The "educational" feedback in the text, a minor perturbation of utter sixth-grade boyishness, was one result of the negotiation. Another result was an (unfinished) tutorial about how to think about graphs, meant for players who had difficulty.

If you choose the lower-right graph in figure 8.14, you get:

| | | |
|---|---|---|
| You speed through the gate, but you slow down, and get your head blown off by a surface-to-air-head missile!! | *long pause* | Die!! |

Or the top right:

| | | |
|---|---|---|
| Michael gets nervous and fights for the control of the car! You crash! | *long pause* | KABLAM!! |

Later in the game, you are performing fancy motions such as accelerating while going backwards in time. You are "rewarded" with, for example:

| | | |
|---|---|---|
| You speed toward the sun, but you don't slow down fast enough and are turned to . . . | *long pause* | cosmic sloppy joes<br>SIZZLE.............<br>..................<br>...............POP<br>SPLAT |

In the midst of the game comes a substantial change of pace. Having gotten away, into outer space, you are attacked by space aliens and get a chance at a shoot-em-up-style video game. Even that game has four levels of difficulty.

These were the early days of working with Boxer, and I had only my prior experience with other computational systems to compare. All of the students in this class produced wonderful projects, but this one stood out for its technical accomplishments. Compared to my previous experience, it was almost unbelievable. How did they do it? In particular, how did they manage to store and reproduce all those graphs? And how had they managed the embedded video game, especially given the hurried end-of-year schedule?

I began to play detective. I knew these two students well enough to know that they were not geniuses in disguise, but the project was inescapably complex. It was larger then my own average Boxer project. I counted boxes. There were nearly five hundred in the full game and much more than one hundred in the space invaders subgame alone.



I began to inspect their code. It didn't take long to find an important clue. Right in the middle of their program boxes, I found a box that was evidently documentation for a graphing tool! It was fairly well written, hence clearly not a sixth grade construction, so part of the program had been "stolen." Later I confirmed with a graduate student that he had brought a partially constructed graphing tool into the class in order to help Sean and Bob with their project. Adaptability and combinability of the medium had helped them. A prior line of development, by others, entered into this project and became part of it.

Let's not jump too quickly past this point. It is absolutely not a trivial point that students could—and should—appropriate some previously written software seamlessly into their own product. First, they had to understand this other software much more deeply than just to be able to use it. They had to understand its "insides" enough to control it with their own code because, for example, they used none of the user interface to the prior graphing tool. Boxer's structure, pokability, and inspectability contributed here. As I have emphasized, one can count on the fact that no previously written tool ever does exactly what you want. In this case, the tool was designed to show one graph, but Bob and Sean needed to show five at once. In addition, the graphing tool was intended to be used mainly in cases where a process is producing values that will be graphed in real time. They had to make appropriate and pretty substantial changes.

When I looked to see what changes they had actually made, I was in for a surprise. They actually took very little code from the original grapher. Instead, they took *ideas*. In particular, they took the idea of representing graphs as a list of numbers, and they took the basic configuration of graphics box, sprites, and code to control them as a framework for drawing graphs. They also used the basic outline of the grapher in terms of how and when to draw the background pair of axes and labels. Most of the rest of the code was completely new, however, even if they could have reused the old grapher code. In particular, in order to "unpack" the lists of point values that they had used to store graphs, they used an inefficient collection of variables, one for each point. The code to draw the graphs themselves was an inelegant, unstructured, repetitious poster child for bad programming style, but it did exactly what they wanted, and it was their creation. What the project lacked in elegance, it excelled in showing organic growth; the grapher was not just pasted into their project, but modified and mutated in ways that matched the needs, interests, and capabilities of the current owners of the line of development.

Taking ideas instead of code is a wonderful advertisement for what a computational medium stands for. It is not simply having a computer with a lot of stuff on it, even stuff that you can reuse. Reusing is fine, but finding and *using ideas* is a much more general accomplishment. It's teaching fishing instead of handing out fish. The fact that the medium expressed ideas for how to do things well enough that sixth-grade students could take the idea, not just the code, goes well beyond my experience with prior media and presents a very hopeful sign. Surely, with a full learning culture (rather than merely the first Boxerized elementary school classroom) to create things and also powerful ideas that can fit this new medium, the scale of these students' accomplishment could be met by many if not most children.

Let me elaborate just a bit on what is happening here. One of the persistent difficulties that I and others had in trying to bootstrap computationally literate cultures was that individual constructions almost always remained individual, so, for example, what one person did in a class almost never propagated to other students. Failure to propagate happened in many, many ways. In groups of students who worked together on a project, one student almost always took over the programming, and the other students became marginal participants who frequently couldn't even effectively use the programming work created by the "group." I had certainly noticed and worried about this phenomenon, but I took it to be a cultural issue. Kids this age just don't collaborate very well, I said to myself, or the technically competent students are being isolationist, which matches our stereotypes of "nerdy" programmers.

Henri had noticed a similar phenomenon in his work. He had long been interested in the idea of computational tools for learning, but his experience was that students merely used the tools, they never took them apart, combined, extended, or modified them, even in simple ways.

Starting to use Boxer was almost like flipping a switch. All of a sudden, collaboration among students improved. It became instantly very rare that one student took over the programming in a collaborative group. To be more precise, it continued to be true that individual students often fell into the role of chief programmer, but this no longer marginalized the other students. Instead, it became the norm that all could operate a group software project. More impressive, we have videotaped exit interviews of students who seemed peripheral in the production of some software, but they turned out completely fluent with its use. They could explain how the program worked and even debug and change the program on the fly. Another surprising occurrence was that during extended group projects, when the "chief programmer" was missing, someone else almost always just sat down to fill in. We still had chief programmers, but with Boxer that role was more a convenience or an efficiency issue, not a necessity. Notice the subtlety of the change from pre- to postcomputational medium. The social phenomenon of chief programmer persisted, but its characteristics changed radically. In particular, having a chief programmer became benign and not a danger to group ownership or group learning. We learned (if we needed the lesson) that technology can have critical effects in areas that we might think are autonomously social or cultural.



Why this effect? How does Boxer facilitate collaboration? I've talked about adaptability and extendibility, but how do *they* work? I believe the core issue is visual expressiveness. The computer screen simply says much more and more effectively with Boxer than it has with Boxer's precursors. The principle of spatial metaphor has a significant role. Space is much more extensively used in Boxer to express meaning than in previous systems, which manifests itself with individuals, but also with groups. In fact, it is very nearly the same phenomenon with individuals and groups. Ted and others who managed complex systems on their own did so because they could effectively collaborate *with themselves*. That is, they could understand and extend what they had done previously in order to continue much farther than they could with a more primitive medium. In groups, this phenomenology becomes evident and explicit. Students are constantly looking at various parts of a complex system, pointing and explaining as well as poking and watching. Much more happens on the screen, and it happens via a richer, more expressive communication channel.

There are other reasons for the better collaboration effect, as well. One worth mentioning here also follows from a principle I mentioned in chapter 7: the principle of computational structuring. Compared to other systems, Boxer shows its basic structures more in the surface of applications written in it. Almost everyone uses data boxes, the ability to execute commands by clicking on them, and other similar resources in the "user interface" to their applications. It's a matter of ease and convenience. Computational structures serve interface purposes, so why construct interface structures that hide the computational structure? The advantage of using everyday Boxer as the interface to programs comes down to learning. Just by using things made in Boxer, our students keep learning about Boxer, so they naturally come to be able to understand, use, and even build and modify programs better. You can think of this process as an instance of the principle of menial utility. Just as well, we can return to the roller-coaster metaphor. In this case, almost any hill whatsoever bootstraps more competence with the medium.

I want to turn toward activity aspects of Sean and Bob's creation, although this will bring us quickly back to technical issues. I troubled to reproduce some of the text from the game in order to emphasize the personal meaning the project had for these students. If you can't imagine two preteen boys cackling and congratulating each other on coming up with an even more gruesome method of dying, I believe you have missed a not uncommon if also not terribly laudable component of young boy culture. Surely "designing death" was a significant part of establishing this project as valued and natural within their existing activity fabric. (I hate to admit it, but I kind of like the cosmic sloppy Joes idea.) What they did was also squarely within their regime of competence. You can imagine how much more teacher pleasing the project would have been without this "wasteful and off-topic" component of the activity. Imagine how much more useful the project would have been as a learning experience had they used all the time they spent inventing disgusting ways to die in order to think more carefully about how to teach graphing or to include more advanced topics, and so on. That is not a choice we have, however. We can accept continuity in the fabric of activity, we can accept off-task but personally meaningful components, or we can settle for no such project at all. Outside the regime of competence, outside a continuity



in the fabric of activity, there would have been no commitment to this project.

Don't mistake this analysis for "let boys be boys" or "let children be children." Like knowledge, we should have goals and expect progress along the dimension of activity, but you cannot jump too far ahead. Their teacher, Tina, pushed and nudged this project toward educational ends, but not so hard that she destroyed it as an enthusiastic personal experience for Sean and Bob. Someday they may have enthusiastic personal experiences that mainly rather than peripherally build learning materials for others. However, that is not where they are in this snapshot. The lesson is that we negotiate, probe, try to understand, and foster possible lines of development. We don't abandon either educational goals or the essence of children.

The space invaders subgame is a relevant case in point. Sean had started this game much earlier in the year as part of another project, but Tina had thrown it out of class. Sean, at that point, could not articulate any connection at all between the game and what he was supposed to be learning in this class, so Tina requested he change topics. Typically, Sean did not abandon his space invaders game. Instead, he was one of the students in the class who requested extra time working on Boxer after school, when he finished his game as a stand-alone system. Sean managed to sneak the game back into classwork in the context of an overtly educational game—graphing adventure. I don't know what negotiation or subterfuge led to that, but it seems to me completely plausible to argue that, as motivation, space invaders served a very useful part of the graphing adventure game—even if there was no direct conceptual connection. The suppression and reentry of space invaders into the motion class represent for me the persistence of activity issues in our pursuit of intellectual advancement.

Technically, how did space invaders enter the graphing adventure game? Issues of medium are again salient. Boxer allows an incredibly simple means of joining programs. Just cut and paste one program into the middle of the other. The spatial metaphor is responsible for this easy and effective strategy. One can always manufacture more space: just insert a new box. When the new box is closed, you have, very nearly, your old world. When opened, you find a new universe in the old place. The contrast case is a typical application in which you simply cannot make any space anywhere to put a new thing without delving deep into complex and invisible processes that make things appear on the screen.

Sean and Bob, in fact, simply pasted space invaders right into the middle of the graphing adventure main box. The game was then instantly playable in the context of graphing adventure; modular, visible box-chunks are wonderfully portable. Then, step by step, they integrated the new box better and better into the game play of graphing adventure. For example, they discovered an advanced feature of Boxer that allowed them to lock the box, keeping players out until they had progressed sufficiently far with graphing puzzles. Then they added features such as having the player's score in space invaders affect the text he would see after he returned to graphing adventure. This is prototypical of organic growth. (1) There is a completely trivial principle of combining things—"paste it in"—even if it is not ideal. No threshold. (2) Then you have lots of time gradually to enhance integration. No ceiling.

## Review

The message of this chapter has been a simple one: children are smart; people are smart. This simple message is not simple-minded, however. People are not perfectly smart in all possible ways, in all possible contexts. Instead, we need to be clever in order to see exactly where human intelligence lies and how to bring it out. In this chapter, we have seen that children possess a remarkable ability to design and think about representations (inventing graphing). Who would have thought that? People also possess powerful abilities to perceive and think about motion (Carol and Ming). Motion competence may seem obvious after the fact, but almost no current instruction of physics is based on exactly what students come into class capable of doing. Instead, instruction in motion is delayed a half decade or more in students' lives while we build an alternative route to understanding motion that relies on more formal means than their natural talents. Children are capable of remarkably large and complex constructions in computational media (Ted's molecular toolkit, Sean and Bob's graphing adventure), but not any computer system allows students to collaborate with others (and with themselves!) in an organic, long-term process of evolving such complex products.



These three classes of competence—metarepresentation, motion, and programming—have very special relationships with computational media. Programming, I have argued from the beginning, transforms limited consumer literacies into more powerful two-way literacies. Motion, along with an enriched spatial expressiveness and interaction, is a fundamental improvement of static media realizable with computational systems. Together, these new modes of expressiveness promise new implementations of material intelligence for humans.

Metarepresentation is the most subtle and unfamiliar of the competencies that we have seen blossom between children and computers given a sufficient computational medium, but metarepresentation may have radical and transformative implications. We may see tool-rich cultures of representational innovators in computationally literate students, teachers, and educational developers of the future. Genres such as the computational essay and the knowledge space may become commonplace in school learning of science. These genres represent not just a new possibility, but a new kind of possibility. Text has limited capabilities as a metamedium, so that representational innovation is minimized a priori.

Although I have concentrated on knowledge, activity is a critical, complementary perspective. Indeed, I could almost as well have called this chapter "Kids Are Engaged," with the same caveats that engagement is culturally achieved—not automatic, not creatable on any ground, and not uniformly supported by any medium. Every story of accomplishment that showed surprising intelligence in this chapter is also a story of dedication and committed learning.