Thoughts About Teaching Science and Mathematics To Young Children

Alan Kay

VPRI Memo M-2007-003-a

Viewpoints Research Institute, 1209 Grand Central Avenue, Glendale, CA 91201 t: (818) 332-3001 f: (818) 244-9761

Viewpoints Research Institute Memo

Thoughts About Teaching Science and Mathematics To Young Children

by Alan Kay

When teaching science, we have the important conflict between the empirical and model-based approach of modern science and the belief-based dogma of religion. This is particularly critical because there are too many important results in science for each and every one to be learned through personal experiment, and many require tools that are beyond most individuals and schools. So, most people including scientists contact science via reading and hearing about results rather than verifying the experiments themselves. Much worse, most children are taught science as a kind of hearsay catechism of "this is true and must be believed".

Scientists escape to a large extent from simple belief by having done enough real experimentation, modeling building using mathematics that suggests new experiments, etc., to realize that science is more like map-making for real navigators than bible-making: IOW, the maps need to be as accurate as possible with annotations for errors and kinds of measurements, done by competent map-makers rather than story tellers, and they are always subject to improvement and rediscovery: they never completely represent the territory they are trying to map, etc.

Many of us who having been learning how to help children become scientists (that is to be able to think and act as scientists some of the time) have gathered evidence which shows that helping children actually do real science at the earliest possible ages is the best known way to help them move from simple beliefs in dogma to the more skeptical, empirically derived models of science.

We also know from the last century of careful observation of children that they think in ways that are different from adults, and as Piaget pointed out, it is best to think of children as thinking beings in their own right rather than as "defective adults who have to be fixed by education". And, further, children themselves are not homogenous in their approaches to life and learning: they have different styles, reasons for why they want to do things, what kinds of play they prefer, and so forth. The adults in their culture are both explicitly and implicitly trying to get the children to be part of the culture, and the children are wired by nature to try to learn whatever this is. In our culture, this is quite confusing, since the adult culture is an admixture of many different approaches to the world and how it works and can be manipulated. Real science is in the back seat or not even in the cultural vehicle as far as most children and adults (including most teachers) and most of the surrounding media are concerned.

My main concern over the years is how to help children and adults do the initial "real science" that can form the modern scientific stance towards knowledge that allows them to deal with the majority of science knowledge presented as claims they will encounter over the years.

So, I'm very interested in how the children can be motivated and helped to observe nature in ways that give rise to the formation of guesses that can be modeled and compared against the observations and lead to further observations. This requires finding out: what motivates different kinds of children, what kinds of observations can be done and in what form, how children can do real modeling and mapping, etc. All this has to be done above thresholds for what "real" means for science and its modeling. These thresholds can be approached by analogies to what it means for children to do "real music", "real art", "real writing and reading", "real sports", etc.

Seymour Papert's background was in real mathematics and science, and he was able to combine these with important insights of Piaget to realize that children could learn certain kinds of powerful math quite readily, whereas other forms of mathematics would be quite difficult. A central realization was that (a) the differential geometry of vectors (Gauss was one of the parents of this perspective) fit very readily into how children thought about themselves in the world, and that (b) the computer could show this world graphically and also easily do the laborious integration of the differential equations to give children the deep hit of fundamental powerful ideas of vector calculus in forms they could recognize and use for their own ends. I still regard these insights by Seymour as among the top few of all time regarding what computers are really good for.

My contribution to this was small, and amounted to adding in the ideas (a) that multiple independent objects (an idea derived from the early world of simulation) which children could program in the manner of LOGO would amount to a world of real children's mathematics that could model many kinds of ideas, including those of science (and of course all previous computer structures and old and new media), and (b) that the entire environment including the math/programming languages the children used were properly part of the user interface and had to be carefully designed.

There is abundant evidence that helping children move from human built-in heuristics and the commonsense of their local culture to the "uncommonsense" and heuristic thinking of science, math, etc., is best done at the earliest possible ages. This presents many difficulties ranging from understanding how young children think to the very real problem that "the younger the children, the more adept need to be their mentors (and the opposite is more often the case)".

If "children first!" is the rallying cry, then it makes sense to try to invent computer environments that use the very best ideas (and these are very hard to come up with). This is why the various groups that got interested in this romantic quest via early contact with Seymour have always been colleagues and never rivals. The hard to come by ideas for projects, representations, user interfaces, experiments, etc., have been freely traded back and forth. The notions of "thresholds below which is not worth going" have been jointly refined, etc. One of the parasitic difficulties is that computer environments, once made (with lots of effort and dedication) tend to form tribal bonds that are rather religious in nature. The amount of effort required plus the attendant religion makes it extremely difficult to take new insights and ideas and make brand new better environments for the children. The strong tendency is to use and reuse and incrementally expand the old environments.

So, for young and youngish children (say from 4 to 12) we still have a whole world of design problems. For one thing, this is not an homogenous group. Cognitively and kinesthetically it is at least two groups (and three groupings is an even better fit). So, we really think of three specially designed and constructed environments here, where each should have graceful ramps into the next one.

The current thresholds exclude many designs, but more than one kind of design could serve. If several designs could be found that serve, then we have a chance to see if the thresholds can be raised. This is why we encourage others to try their own comprehensive environments for children. Most of the historical progress in this area has come from a number of groups using each other's ideas to make better attempts (this is a lot like the way any science is supposed to work). One of the difficulties today is that many of the attempts over the last 15 or so years have been done with too low a sense of threshold and thus start to clog and confuse the real issues.

I think one of the trickiest issues in this kind of design is an analogy to the learning of science itself, and that is "how much should the learners/users have to do by themselves vs. how much should the curriculum/system do for them?" Most computer users have been mostly exposed to "productivity tools" in which as many things as possible have been done for them. The kinds of educational environments we are talking about here are at their best when the learner does the important parts by themselves, and any black or translucent boxes serve only on the side and not at the center of the learning. What is the center and what is the side will shift as the learning progresses, and this has to be accommodated.

OTOH, the extreme build it from scratch approach is not the best way for most minds, especially young ones. The best way seems to be to pick the areas that need to be from scratch and do the best job possible to make all difficulties be important ones whose overcoming is the whole point of the educational process (this is in direct analogy to how sports and music are taught -- the desire is to facilitate a real change for the better, and this can be honestly difficult for the learner).

The detailed parts of the design have to do with (a) what kinds of math and science the children can do and learn, and then (b) by good solutions to the user interface and expressive elements in the computer environment. These co-evolve because certain things the children can do and learn only have a real payoff for them if they have a computer. For example, many areas of physical dynamics (e.g Galilean Gravity) can be explored and represented in a kind of differential model they can understand without using a computer at all. But for all but about 5%-8% of the kids just finding a good model is not enough of a payoff. However, if the model can be set into motion (in mathematical terms: the integration of the differential model) then many "pieces of art" of great interest to about 90% of the children can be easily made. These include various kinds of falling games (like Lunar Lander, shoot the monkey, etc.).

As the children get more sophisticated, the black and gray boxes that scaffold what they are doing can be popped open and understand and modified. For example, "forward" (which moves an object in the direction of its "heading") is a black box initially, and very useful in that form. But there is a point when the children will be greatly aided by understanding that forward is just a vector addition and is a method made from a more fundamental idea. The underlying language for the system itself has to reveal itself as the same species as what the children have been learning. The analogy to how English is carefully used for different ages and the expanding range of ideas and expression is quite apt.

One of the simplest rules of thumb for any kind of design that requires learning is George Miller's "7 plus or minus 2", which refers to an estimate of how many things can be given attention at one time by our limited human minds. Perspective, simplification, abstraction and duration can all be traded off in designs that try to help learners make progress without overwhelming them. In programming language design in a UI, especially for beginners, this is especially crucial because it is easy to go far beyond 9 new elements. Many users will interpret this as "I am stupid and can't do this" rather than the more correct "The UI and language designers are stupid and they can't do this".

It's hard to point to any programming language for beginners that has a really great form. One thing that has consistently worked is "close to natural language but clearly not natural language". That is, it really helps if the gist-view of a program is a kind of metaphor for what it does, even if one has to think harder about the detailed meaning. For children, Hypercard was OK in many respects for the gist-view, but was too like English for both deep understanding and for programming (many children had a hard time getting past the idea that Hypercard couldn't understand and do any reasonable English sentence). This was debated endlessly in Logo circles, and Logo wound up going from a much more English-like syntax to one much more like Lisp (this was a big mistake in my view). Finding the balance between these is critical, because it governs how much brain is left to the learner to think about content rather than form. And for most learners, it is the initial experiences that make the difference for whether they want to dive in or try to avoid future encounters.

In one respect, young children are easier to design for in that there are fewer standards for math and science in the early grades (this is being eroded, especially wrt math). If we take functional relationships as an example, it has been shown that children readily understand them but have considerable difficulty with variables, and much more difficulty with parameters. The standard math syntax for functions with parameters requires some extra trained chunks to associated dummy names with actual parameters. Some computer languages allow conventions for prefixing the actual parameters with the dummy names. This is good. For younger children, it's likely that making these into complete assignment statements is an even better idea. An object oriented language can use instance variables for a long time before introducing the idea of passing parameters in a method, etc. Having really good trace and single-step visualizations is critical.

The importance of gisting argues against forms that have more than one meaning. For example, many languages (going all the way back to the 50s) have decided that "=" should have a double (or triple) meaning, and that it is up to the programmers to simply deal with it. This is almost certainly a bad idea for children K-8 (and beyond). In the case of "=", even in a functional language, the difference is huge (between a functional relationship and a definition). A suitable "logical language" that used unification would be less ambiguous but introduces quite a bit of abstraction (likely too early).

But, as the child gets closer to high school and the outside world, the QWERTY phenomenon starts to get more important. Many of the global standards are willy nilly and poor, but need to be pragmatically learned at some point. For example, it is relatively easy to get 10-11 year old children to do the real science and math to make a good simulation model of the 2nd order differential relationship. This is best done (so far) by using two accumulator addition (used to be called a DDA and goes back before Babbage). The trade-offs between this simple and deep way to think incrementally about what is going on and the standard algebraic way of thinking about "what's going on for all time all at once" are interesting and important.

Our view is that one should get the 5th graders to do this in the most powerful way for them (incremental addition). And then later (maybe in 8th grade) revisit this and see how the incremental approach can be turned into an algebraic perspective. Both ways of looking at this are important and powerful in their own ways. It is likely that educators are quite justified in making up useful non-standard representations for powerful ideas for any topic that would be helped by this for children below the age of 12. There are many other important reasons for introducing more powerful forms for dealing with ideas in early childhood, etc.

So it seems to me that there is a lot of room for new and different ideas for children's environments for learning powerful ideas. They have to be above threshold and in the spirit of real science and mathematics. Two cautionary examples are Interactive Physics and SimCity.

The first assumes that Newton was absolutely right and is a direct embodiment of Newtonian Dynamics and Cosmology. The users are restricted to paramerizing the internal dynamic models and cannot see them, question them, or change them. (For example, it is really important to be able to try an inverse cube law for gravity, etc.). This is most assuredly not in the spirit of science! It amounts to a dynamic bible. In order for this to be useful in real education, there has to be a lead up that derives the relationships in an empirical and mathematical form, and only then will the premises of IP be useful.

SimCity is similar but more pernicious. It is a black box of "soft somewhat arbitrary knowledge" that the children can't look at, question or change. For example, SC gets the players to discover that the way to counter rising crime is to put in more police stations. Most anthropologists, sociologists, psychologists, and economists would disagree violently. Alternate assumptions can't be tried, etc.

Both of these packages have won many "educational awards" from the pop culture, but in many ways they are anti-real-education because they miss what modern knowledge and thinking and epistemology are all about. This is why being "above threshold" and really understanding what this means is the deep key to making modern curricula and computer environments that will really help children lift themselves.