What's next?

Deborah Tatar

I purely love the idea of resuscitating/continuing Boxer. Boxer was an exciting project in the early 1980's in my experiences at M.I.T., then during its nano-second appearance at Le Centre Mondiale de Informatiques et Resources Humaines in Paris, and then when we tried to implement it at DEC as a mechanism for educational computing integrated with the then-novel technology of video on CD-ROMs. It's still exciting today, for two reasons. The happy reason has to do with its nature and design: because it embodies the promise of integrating literacies, because it is careful, careful, careful in the ways it presents metaphors and enactments of computing, because it integrates exposure to the few truly difficult elements of computational representation and process with the more easily obtainable elements. It promises democratization of education and access by small increments for everyone to the most important tools to think with. But it is also important for an unhappy reason. The unhappy reason is that many other attempts to introduce low-threshold/high-ceiling instruction have not worked. Boxer stands along with a tiny cadre of other tools and systems like a mesa that has survived when all the lesser rocks have been washed away. How appropriate that Andy was born in Colorado. 40 years is an eternity in educational reform. I think that what gives this rock its endurance is that the underlying inquiry has to do with what is truly hard to learn. As Piaget long ago observed, most learning is simply assimilation to an existing schema. But some really requires a shift. Changing the representational infrastructure can truly change access to thoughts that are otherwise hard to think.

A few other projects stand out as opening up reformation of the representational underpinning for learning. Ones that stick in my mind are (in no particular order) Logo, SimCalc, Media Computation, NetLogo, Scratch, AgentSheets, Hypergami, Sewable Computing/Lilypad Arduinos, Line Museaus' work on integrating modeling into biotech instruction, and my own two integrated Computational Thinking projects including ChemC. Most people will not have heard of any of these, not even Logo and Scratch, which have had the wider popularity.

There are four elements to the lack of impact.

One: Low-threshold, high-ceiling educational technologies, technologies that change what we offer children in classrooms to think with, pretty much violate every one of the factors that E. Rogers identified as critical in the diffusion of innovation: clear relative advantage, high compatibility, low complexity, easy trialability, and observability.

Two: The people who need them most want and understand them the least. I spent ten years examining whether SimCalc could indeed democratize education the United States. The answer is "Yes. But." Otherwise known as "It could but it won't." Amazingly SimCalc can lessen the gap between algebra learners in low socio-economic settings and higher ones. Pause for appreciation, because almost nothing does that. But SimCalc is more effectively utilized in high SES setting because teachers in higher SES settings are better trained, have easier times with classroom management, are paid more, and consequently have a different vision of instruction. SimCalc is not easy to teach with, but rather hard. So, it does not eliminate the gap but does give children in low-SES settings more of a fighting chance at learning algebra (and therefore attending a four-year college) because it exposes them to a kind of thinking that would otherwise be invisible.

Three: Educational systems are truly complex and unhappily political. When we need gentle rains, we have gully-washers. The nuance to see how a brilliant educational intervention can operate (with hard work) does not exist in a world that cannot adopt something as simple as masks or vaccinations that clearly save people's lives. It does not work in a world in which people confuse reading with knowing the ABC's and math with arithmetic calculations. You can't fool all of the people all of the time, but maybe you can fool enough of them enough of the time to (for example) destroy the planet. Today's news reports on a long-term study in Tennessee that showed that children that had pre-school performed worse than children who did not ((https://www.npr.org/2022/02/10/1079406041/researcher-says-rethink-prek-preschool-prekindergarten). It is completely obvious that the problem was not pre-school itself, but how the people in Tennessee conceived of pre-school. But instead of making their pre-schools better, funding will be taken away from the educational system.

Four: Systems have unintended consequences. In particular, low-threshold systems have a long history of appropriation. The very features that make them low-threshold mean that they can be used by younger and younger children. But neither those same children nor their teachers are ready to plumb the implications. Thus Logo, or Media Computation involve a certain amount of intellectual candy that can be offered to fifth graders, who appreciate it. But, having having given it to the fifth graders, and then to eight graders and then to tenth graders, you cannot use that same candy to draw college students in to the more complex aspects of computing. Years ago, I wrote a paper with a subtitle "How standards become the upper bound". Often when these elements are taught as stand alone bits to young children, they operate to close the world down rather than opening it up.

This is a rant. My apologies. But it is a placeholder for the piece that Aakash Guatam and I would like to write about the relationship between teaching about ill-structured and well-structured systems; abstraction, decomposition and evaluation in sciences vs. computation; and how each of these relates to top-down thinking vs. bricolage.