Five Powerful Ideas about Technology and Education

Andrea A. diSessa
Graduate School of Education
University of California, Berkeley, USA
disessa@berkeley.edu

Abstract. *I present "five powerful ideas" concerning the very best ways to enhance education via technology. These are "big" ideas that have guided decades of my own work. They have been reinforced and adjusted with experience. However, these ideas are subtle and in some ways cut against the grain of popular trends in thinking about technology and education. Therefore, progress in implementing them will require careful thought and public engagement.*

Keywords. Computational Media, New Literacies, Re-Mediation, Toolsets.

1. Introduction

I aim here to distil my four decades of experience with technology and education into just a few ideas. Many of these ideas are absent from trends in popular discussion. Indeed, they sometimes run in the opposite direction. While discussing these ideas, I will point out some dissonances with "contemporary" thought, along with some more positive connection. All of these ideas are complex and subtle. So, interested readers should consult other sources, such as [1].

"The computer is a once in several centuries innovation." The first thing to take into account in looking for profound directions is that they will not be easy or short-term efforts. If they reach fruition, however, they will constitute grand cultural achievements, something our civilization may take pride in reaching.

2. Idea One: Computational Media and New Literacies

I think of computers as providing the basis for a new literacy. Literacies are big deals. They take many years, sometimes centuries, to spread widely and have their deepest effects. Textual literacy was clearly a big deal in the history of civilization. Many have studied the long paths to achieving widespread literacies, and no one doubts that they have had monumental transformative effects. Societies just work differently when they have both history and laws, which, for all practical purposes, cannot exist without textual literacy. Science, also, is essentially a literate pursuit.

Computers can offer easy-to-understand extensions to the power of textual literacy. Text is static in two senses: It does not change, and it does not respond interactively. In contrast, computational media are essentially dynamic and interactive. In expanding to include dynamics and interactions, they can engage intelligences (including our powerful spatial interpretive and imaginative capabilities) that are near-dormant with conventional literacy. These are not necessarily things that draw our attention. It is good to remember how boring (in a visual sense) conventional text is. And yet it made our modern world possible. New media's modes of helping us engage and extend ways of thinking that lie deep within us—intuitive, imagistic, and enactive thinking—are nearly untouched by text.

Conventional literacy sets a proper scale. A new literacy would be a grand cultural achievement. As such, it lies beyond "improving" our current ways of doing things. It will establish modes of thinking and interacting quite unlike what exist now. Realizing the promise will be challenging in many senses, including that few people are conceptualizing or working toward these changes.

Computational media are expressively unlike text. Text is broadly applicable, but also imprecise and not easily adapted to specialized niches, like science or mathematics. A good comparison is to algebra as a literacy specially adapted to mathematics and science. It brings huge increments of precision and relevant expressiveness. Indeed, the history of algebra is instructive. While it began in the mid eighteenth century, it was only in the twentieth century that algebra became a widespread literacy, essential for any technical trade or profession, expected to be learned by everyone. Algebra is particularly useful as a comparison because it engages modes of thought that "don't fit" easily in text, but which (quantitative thinking) can be hugely expanded in efficiency, precision, and coverage, compared to text. Like text, algebra has become infrastructural in our educational system. Everyone is expected to pass through algebra on the way to college and beyond.

The expressive range of computational media is huge compared to algebra. Over the long haul, it will have a much bigger and broader effect, especially in techno-scientific traditions. Computational media are also much easier to learn than algebra, and they extend into areas with which "ordinary folks" are interested and more competent than those approached by algebra: visual art, interactive story telling, "computer" and social game construction. In our work, students' affinity toward computational media for their own interests has been transparently evident.

The histories of other literacies follow patterns that may be repeated with computational media. Literacies develop slowly and often without notice. During development, few conceptualize the ultimate pervasiveness of the literacy, the depth of its possible influence on thinking, and its civilization-wide impact. Instead, literacies are first conceptualized as technical, and encapsulated in exotic professions, such as "scribe." Literacies start as "one-way," something for many to consume (reading the thoughts of the master) but few will produce. Yet, one-way literacies, reading without writing, are in the end, impoverished.

There have been few cultural resonances with true literacies in the history of technology and education. Early on (and maybe still today) people construe "literacy" in a denatured sense, as "shallow competence"—something with which someone ought to have some passing acquaintance, not something that is infrastructural to civilization and pervasive in school. In this regard, recent attention to programming and "computational thinking" is heartening. But the movement is in important ways ignorant of the larger possibilities and of the history of those aiming to make "writing" in computational media ("programming") a part of everyone's experience [2]. Advocates of learning programming also often do not see it as infrastructural, relevant to learning other things. The meme of vocationalization ("we need more professional programmers") obscures and marginalizes the big picture of new literacies.

3. Idea Two: Re-Mediation

Thinking is symbiotically enhanced by all the external representations that humans have designed for their own intellectual purposes. Computational media will enable a myriad of external representation unrivalled in history, including: variations on text specialized for dynamic human interaction (social media); schematic and realistic (or real) still and dynamic pictures; all the specialized representations that have already extended familiar ones, such as nimbly adjustable extensions of "graphs" in exploratory data analysis. Among these representations, a special class are the means to specify action and interaction, the core innovations of computational media. We can call them "programming languages."

One fundamental fact about representations is that each has a delimited expressiveness. They "talk about" certain kinds of things well, and other things poorly. Algebra and calculus were fabulous facilitating representations for basic forms of physical dynamics, like Newton's laws. But, modern science has transcended the classical advantages of such representations. One doesn't predict the weather or explore fusion anymore by solving equations. One builds models using programming languages.

What will happen when computational representations come in contact with mathematics and science education? The easy prediction is that all the new sciences that have computation in their very core—data analysis, complex systems—will become newly feasible targets in school. The more difficult thing to understand is that all the old things that we once taught on the basis of text and static extensions will be changed almost unrecognizably when we "re-mediate" them with new representations.

One of the wonderful early experiments I organized with computational media was to teach sixth grade students high school physics (mechanics) in a yearlong class. The curriculum seemed wildly unrealistic to reviewers of our first proposal. In particular, they picked on the fact that we intended to teach vector formulations of the laws of mechanics. Vectors are now, indeed, a difficult part of high school curriculum; it seemed to reviewers outrageous it might be possible to do this in the sixth grade.

I adjusted our "expectations" to those of the reviewers, and we were funded. However, we did teach our students about vectors. We found it not only successful, but trivial. Vectors were, for these students, simply arrows on the screen that they made with a key-press and could be adjusted with the mouse. The meaning of vectors could be established, for example, with a one-line program that directed a graphical object to move with the velocity specified by a vector. Vectors, thus, became a "direct manipulation" interface to motion, and students could trivially see the effect of changing size or direction of velocity. Dynamic and interactive control over motion not only made the meaning of vector velocity (or acceleration) transparent, but it established a class of activities in which students enthusiastically engaged: making video games using vectors as "control" devices.

So, science and mathematics curricula can be liberated in unprecedented ways with computational media in terms of selection, ordering, and (see below) mode of student engagement. Explorations are just beginning; in the best of circumstances, there are decades of work to reform our educational system to optimally take advantage of re-mediation.

Sadly, this wonderful and dramatic task, exploring re-mediation in many or all school subjects, has hardly been engaged. The problem is worse that an unrecognized or unfunded possibility. The cultural trend toward a standards- and testing-based educational system could not be picking a worse time to legislate details of scope and sequence just when all the old assumptions need no longer apply. Without noticing it, our society may be freezing in constraints based on the affordances of old media, and freezing out perhaps the best possibilities of re-mediated thinking and learning. Standards, just now, are dissonant with achieving the best with new-media literacies.

4. Idea Three: Engagement and Activity Structures

I am struck dumb as to why mathematics and science textbooks are so alike. You read, and then you do problems. Has this mode been established to be optimal?

While I am sure no one knows, and hardly anyone cares, I *might* imagine reading and problem solving may be the best way to learn science using old (static) media. Yet, I am convinced by logic and experience that we can do much, much better with new media.

Our sixth grade students responded to vectors as (1) easy-to-understand and, as important, (2) their pathway to things that truly interested them. They made games and simulations with them as if vectors were sticks and balls from the toy closet. Not only do some things become so much easier with computational media that they can be taught a half decade earlier, but students' mode of engagement might also be radically changed.

Let me inventory a short list of powerful modes of encounter for students that are greatly enhanced by computational media. First, in the case of our sixth grade students much of their learning appeared to them to be game playing or game constructing. There is resonance in the zeitgeist. "Learning through games" is now at a pinnacle of interest among educators. (Much of this energy is unconcerned with conceptual re-mediation. So, educational effects may be mostly transient and hard won, unlike our shockingly easy accomplishments with re-mediated vectors.)

Another mode of engagement with content that is immensely facilitated by computational media is design. In our own work this has taken two somewhat different forms. First, we had children design things that turned out to be proxies for scientific principles. In one case, we asked our sixth graders to design a simulation of dropping a ball [3]. What emerged, with little guidance, was one of Galileo's great accomplishments, a conceptualization of gravitational fall, expressed in a simple program.

In another case we asked students to design a program to simulate a spaceship with a short-burst rocket engine [4]. This was a little trickier and at the high school level. But again, it seeded ideas with computational representations (vectors, unlike the dropped ball model). Students progressed to a good general representation of Newton's conception of the effect of forces on motion in computational form (rather than using algebra or calculus).

The second mode in which we engaged design as a primary form for instruction is all the more pregnant with new possibilities provided by computational media. We asked students, from sixth grade to high school, to design representations suitable for scientific presentation of natural phenomena, from motion (again, in our sixth grade class), to representations of topographical features, to the design of aspects of computational representations of astronomical images. The wonderful synergy here is that, with the explosion of representational resources provided by computational media, we should cultivate students' "meta-representational" capacities to design (and understand the design rationale behind given representations) far beyond what is now in the curriculum. A foundational scientific result of this work is that even children possess a remarkable foundation of ideas—and interest in—representational design [5].

Here, as with learning via games, the zeitgeist seems on the side of re-embedding content in design, or to value design, itself, as a new target of instruction. This is at least true in engineering, where, for example, the state of Massachusetts has mandated (engineering-based) design instruction in K-12. The trend is also vivid at the university level where there are strong currents to engage design much earlier in the curriculum.

A final "new" activity embedding for learning is research. In college-level physics and mathematic courses we designed, students did original research as freshmen, rather than learning by solving a set of artificial, and boring, problems, the universal mode for current algebra-based physics. Similarly, the mathematics we taught surpasses proof- and problem-based instruction, reaching independent student research. A proportion of this work (mostly mathematics) may be found in the textbook we produced [6].

Cultural resonance to research as an activity embedding for science instruction is tenuous. While there is a lot of research interest in activity-based science in K-12, anything resembling actual research (with uncertain outcomes) is rare in schools. There is, however, a fairly strong resonance with attempts to get undergraduates involved in research ("undergraduate research opportunities" programs). Yet, most of these make no contact whatsoever with core instruction in the sciences. An encouraging niche involves recruitment and retention programs that aim to make science a more attractive and meaningful endeavour for all students. Consider courses (such as Phys 98) listed for the excellent Compass Program at UC Berkeley: www.berkeleycompassproject.org

5. Idea Four: Open Toolsets

The maturing of computational media will take a huge number of innovations, both technical and cultural, to realize a powerful, infrastructural literacy. This and the next section provide examples of such innovation. The first is mainly technical, a scheme for software design; the second is primarily cultural, a new social model for educational software production.

Media such as written text or programming languages are generic. Being generic is, in principle, a wonderful thing. Such media are expansive in their application and have the property of "learn once; use forever." That is, any expertise gained with the medium can be used again and again, in whatever context, for whatever mode, for whatever topics or purposes. For our experiments with computational media and new literacies, we developed an environment that is, in essence, a fusion of a programing language and a hypertext processor. The system is called Boxer, and our design intended to do at least as well as text and conventional graphics in terms of static media, but extended those capacities with constructible and reconstructible dynamic and interactive resources [7].

However, generic media have a critical shortcoming. The distance to specific application may sometimes be too large to suffer. There may be what some call the "Turing Tarpit," where everything is possible, but nothing is easy. A more apt description is that some things may be easy, but few of them are exactly what you want to do.

To bridge the gap, we developed the idea of open toolsets. The idea is simple. A particular domain can be approached by building a set of tools that are adapted to the domain, and yet have the following properties: (1) They are built using the generic resources of the medium, hence anyone can, in principle, open them up to see how they work, or change them. (2) They appear in the system as generic objects. In Boxer, every object is a "box," so most tools are just boxes that can be cut, copied, pasted, or "opened" to reveal their insides, how they are constructed. (3) Tools may be easily combined using generic resources of the medium. In the simplest case, multiple tools can be used together simply by copying and pasting them together in the same place. In more complex cases, tools can be programmed "from the outside" by sending them messages that are nothing but programming commands. Or, gestures can be used to interconnect tools, such as "wiring" them together in the way that electronic devices are constructed by wiring together components.

The vectors mentioned earlier are a very simple Boxer open toolset. Vectors are ordinary Boxer graphical boxes that show a vector as an arrow, and allow one to drag the vector's end-point around with a mouse. To manipulate vectors with programs, we also added commands to the language, in the usual Boxer way, that allowed simple expressions to, for example, add vectors as one conventionally adds numbers. Finally, we added other simple commands to allow vectors to interact with generic graphics boxes, for example, commanding a graphical object to move as a vector indicates—displacing in the direction and with the magnitude of the vector. Moving with the speed indicated by a vector is a one-line program using these resources.

Another very successful toolset that we built was designed to allow anyone, curriculum developers, teachers, or students to play with and build constructions that do image processing. Our main application was to astronomy, in particular processing images of the heavens to allow analysis of stars, planets and galaxies, just as astronomers do. [8] explains elements of this toolkit and richly describes the uses that both we and students made of the toolkit. For our part, we (as ersatz curriculum developers and teachers) were able to build, very quickly and on the fly, exercises and activities for students: (a) to explore image processing in general, (b) to use image processing to aid in discovering and exploring visual phenomena in general, and (c) to conduct astronomical investigations using images from telescopes. Constructing an exercise or exploratory microworld for students was often a matter of an hour or so of work, using our toolset, and it seldom required as much as a day of work.

The flexibility of this toolset also allowed students to move off on their own to explore or play with a variety of things that were of interest to them. One student, for example, used some of the tools in the image processing tool set to explore the construction of beautiful palettes of colors with which to process images for aesthetic effect, like Photoshop or Instagram filters.

Other toolsets that we designed [9] included some to explore evolution, plant growth, ecological processes (population dynamics), and databases to allow flexible querying of relatively large data sets, or even just to store useful data for menial purposes. We even built a toolset for easily constructing specialized tools for the analysis of video data in our own research.

In short, open toolsets provide resources specialized to various domains and educational resources, but they do not restrict developers, teachers, or students in their own use of those tools.

6. Idea Five: The LaDDER Model

Literacies are ideological. They embody—or at least uses of them embody—orientations that are characteristic of communities or cultures. For example, the Internet in most people's eyes embodies openness and democratic principles. Similarly, the kind of media and literacies that my group has espoused are strongly democratic. We want everyone to have access and capability to use all the resources of computational media. Especially with technology, power and capability tend to reside at the "top," and narrowly in technically adept subcultures.

Fig. 1 illustrates a mode of creating software with which we have experimented. It is called Layered Distributed Development of Educational Resources (LaDDER). The point is to push competence and capacity toward the lower, traditionally less technologically privileged levels. In this figure we stop with "teachers." In general, we most certainly would want to include students.

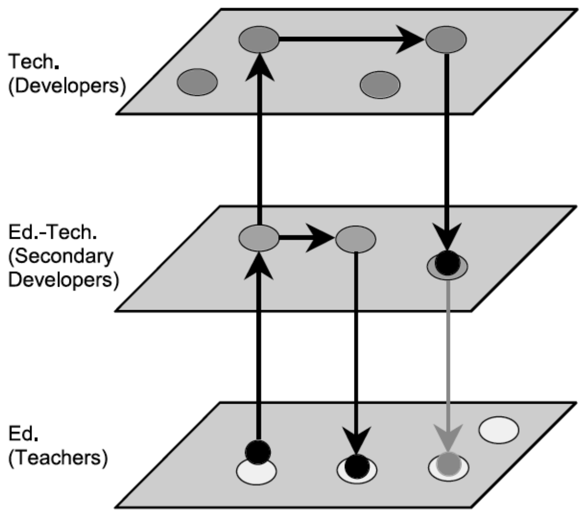

Figure 1. The LaDDER model

In Fig. 1, problems or gaps in capacity appear as black dots. Those problems are percolated upward (upward arrows) until they can be solved. However, the best solutions are not solutions, per se; they should be new resources or know-how that can be percolated back down the layers (downward arrows) so that, ideally, everyone can solve not only the problem that initiated the process, but related problems as well.

We experimented with the LaDDER model in collaboration with a school district in Florida. Originally, a technologically experienced and university based mathematics and science specialist came to us asking for a little tool to create colored number charts (for example, coloring all the multiples of 2 in blue on a 10x10 chart of the numbers from 1 to 100). This is a familiar form in traditional textual (printed) form around the world. But this specialist believed that a flexible, interactive form could achieve far more than traditional forms.

Over the next few years, working closely with local teachers, this specialist developed an extensive curriculum for elementary math on the basis of the original toolset and extensions. Each summer, he would return to Berkeley for a visit with a wish list of items, some from him, some from his teachers. For example, he wanted to open aesthetic avenues, not just mathematical ones, so he wanted the simple capability not to display numbers in the colored charts. We obliged, but also suggested that he use some of the color and palette generating tools that we originally developed for image processing. His collaborating teachers wanted to change the interactivity of the chart so that students' clicking on the chart could be interpreted as the answers to questions in little interactive quizzes they produced for students. We obliged with "hooks" so that interaction could be modified in general and at will.

Eventually, the curriculum was extensively tested in a large scale, random assignment study [10]. It was impressively successful, and we believe (but cannot prove) that a part of the success was building deep attachments to classroom practice via involving both this math specialist and teachers in the creation of suitable software. The educators did the work, and we just helped them with resources they could use to do the things they wanted to do.

7. Conclusion

The promises of technology in education are grand. But, realizing the best will be a subtle, long-term enterprise, far beyond—and different from—what many expect. What draws people's attention to technology is often simply not on the path to the best that we can imagine. This note presents some of the best ideas I have collected and developed in my career as an educational technologist. We should think at the level of new media and computational literacies. We should exploit re-mediation to re-shape the curricular landscape. We can now re-embed learning in activities that students find more personally meaningful, such as design and research. Finally, we should explore flexible new forms of software and social organizations for producing them.

8. References

[1] diSessa A. Changing Minds: Computers, Learning, and Literacy. Cambridge, MA: MIT Press; 2001.

[2] Grover S, Pea R. Computational Thinking in K–12: A Review of the State of the Field. Educational Researcher 2013; 42(1): 38–43.

[3] diSessa A. Can Students Re-invent Fundamental Scientific Principles?: Evaluating the Promise of New-Media Literacies. In: Willoughby T, & E. Wood E, editors. Children's Learning in a Digital World. Oxford, UK: Blackwell Publishing; 2008. p. 218-248.

[4] diSessa A. (1995). Designing Newton's Laws: Patterns of Social and Representational Feedback in a Learning Task. In: Beun R-J, Baker M, Reiner M, editors. Dialogue and Interaction: Modeling Interaction in Intelligent Tutoring Systems. Berlin: Springer-Verlag; 1995. p. 105-122.

[5] diSessa A. Meta-representation: Native Competence and Targets for Instruction. Cognition and Instruction 2004; 22(3), 293-331.

[6] Abelson H, diSessa, A. A. (1981). Turtle Geometry: The Computer as a Medium for Exploring Mathematics. Cambridge, MA: MIT Press; 1981.

[7] diSessa A, Abelson H, Ploger D. An Overview of Boxer. Journal of Mathematical Behavior 1991; 10(1), 3-15.

[8] Azevedo F. Personal Excursions: Investigating the Dynamics of Student Engagement. International Journal of Computers for Mathematical Learning 2006; 11(1), 57-98.

[9] diSessa A. Open Toolsets: New Ends and New Means in Learning Mathematics and Science with Computers. In: Pehkonen E, editor. Proceedings of the 21st Conference of the International Group for the Psychology of Mathematics Education, Vol. 1; 1997. Lahti, Finland: PME 21, p. 47-62.

[10] Ploger D, Hecht S. Enhancing Children's Conceptual Understanding of Mathematics Through Chartworld Software. Journal of Research in Childhood Education 2009; 23(3), 267-277.