On the Nature of Literacies and Literacy Development

Andrea A. diSessa
University of California at Berkeley

My comments here will be drawn largely from directions instigated by Brent Davis's commentary on my talk. Those directions are plenty rich and important enough that I won't bother second-guessing in order to find "the optimal thing to discuss." I consider his comments to be offerings aimed to make computational literacy more comprehensible to a broader audience. Thanks, Brent.

In terms of "housekeeping," however, I would like first to mention that the paper Brent read and referred to was not the one my talk was based on. The conference organizers suggested that I make the paper that he read available—and it's one of my favorites. But I had already given a talk based on that paper to an overlapping audience. So, at the risk of some confusion, we made one paper available, but I talked about a somewhat different set of issues, from a different perspective. A paper more closely aligned with my talk is:

> diSessa, A. A. (2016). Five powerful ideas about technology and education. In M. Riopel & Z. Smyrnaiou (Eds.), *New developments in science and technology education* (pp. 63 – 72). Heidelberg: Springer. Available at https://link.springer.com/chapter/10.1007/978-3-319-22933-1_7

The Nature of Computational Literacy

A lot of Brent's commentary addresses the issue of just what computational literacy can possibly mean. The fact that it seems mysterious to him is not at all beyond my comprehension or expectation. In the book that first introduced the idea in detail (diSessa, 2000), I start by explaining a bit about why it is unfamiliar and seemingly exotic. The short story is that it does not mesh with the widespread cultural memes of the day. So, in pursuing computational literacy, there is a lot of clutter[1] that needs to be moved out of the way, and a pathway to a new direction needs to be seeded and cultivated.

The cultural situation has gotten both a little better and a little worse in the 20 odd years since I published that book. What's gotten much better is the vivid public presence of the meme (culturally wide-spread idea) that programming is something that every student should participate in. When I got started in technology and education, a then-current predominant meme was the reverse: Nobody needs to understand programming any more than they need to understand how their washing machine works (Really! References on request.). What has gotten worse is that the conception of what programming gives to students has been muddled and misdirected (my judgement) by alternative narratives that have become popular. Chief among them is the idea that people just get smarter by engaging in programming, in whatever field they choose to pursue. Other current memes include "you can get a good job in the tech industry this way," and "this is for the good of society; knowledge of what's going on under the hood affords a kind of public oversight to avoid the mayhem that bad-actor technologists can inflict on society." This last civics-oriented meme favors understanding something of the insides of things like algorithms that make recommendations, or order search hits for individuals—or soon, if not already, the perils of AI in the public forum.

[1] Brent's search of the internet for wisdom on what computational literacy means, I'm sorry to say, documents a lot of the cultural "clutter" that mostly obscures better-founded ideas.

I have always thought, and continue to think, that people can self-bootstrap into the concept of computational literacy via their massive experience with textual literacy. It simply is not difficult to see the depth and breadth of effects of literacy on our civilization. Imagine: No "nation of laws, not men"; no history; no scientific paper; no textbooks; no classics of literature. Every modern country is measured by its literacy rate, and national programs to improve literacy are strong, even in countries with high literacy rates, like the U.S. and Canada.

I have also found it easy to bootstrap conversations about whether, for example, algebra/calculus is a literacy, or not. People do understand the power and some of the main characteristics of textual literacy, and against that, they make judgments about whether knowing algebra/calculus meshes. Now, there are many ways to slice this, given that literacy is not a technical term (with attendant clarity of breadth and meaning). Not *everybody* needs algebra/calculus, and even those who do need it don't need it every day, for many different purposes, unlike textual literacy. Algebra/calculus does meet my own technical definition of literacy. (Consult the discussion on "what is a literacy?" in the first chapter of diSessa, 2000.) But, I can definitely see the sensibility in people drawing lines in a different place.

So, here's a core observation. Literacies can come in many, many varieties, at least as different as the many communities that adopt literate practices at their core. What we want to do is not only—or even primarily—decide whether we want to call something a literacy, but instead establish the modes and importances that any literacy-like set of practices entails.[2]

In that spirit, I'd like to lay out some plausible futures in the development of what I'd call computational literacy, suspending any certainty in using the term, opening space for readers to make their own decision. Literacy development is a project of immense uncertainty, which is why I often accede to the observation that "we just don't know what (future) computational literacy will look like." But I simultaneously maintain that there is no doubt that literacy-like things will (or, even more certainly, *can*) come to exist. I'm sorry that we cannot foresee their particular forms more clearly. But such is the fact of diversity in literacy types, and their complex relations to knowledge and social structures.

All of what I say below involves fairly simple extrapolations of things we (myself and my extended community) have experienced already in teaching actual students in actual classrooms, as laid out in my talk. So, this is not pure conjecture, but it is extrapolating what we have actually seen with students and teachers to the level of (potential) societal literacy.

Multi-causal Complex Systems

Before I get going with the nitty gritty, I want to take a minimal stab at characterizing, at a very general level, the nature of literacy, to compare with potential new literacies. I believe literacy is a very complex, *multi-causal* system. "Multi-causal" is important in that the system contains sub-systems with very different properties. Global warming is multi-causal because it depends not only on the dynamics of the atmosphere, but also on human reactions to relevant facts, and even political responsibility and action.

First, and most obviously, there is the technical perspective, which concerns the computational medium, the material substrate (like text) for the literacy. I think it is obvious that a substrate for a true literacy has to have properties that one does not typically measure for things like "programming languages." The first focus I would attend to is simplicity and learnability,

[2] One thing that has gone wrong in discussing things like computational literacies (or computational thinking) is that the focus has been too much on computation, and not enough on literacy practices, the bread and butter of how familiarity with a representational system (like text, or programming) affects community-wide intellectual practices.

over broad age-ranges, for explicitly educational tasks. So, I will assert with great confidence (although, of course, I accept some people will contest it), that the system we have been using, the Boxer computational medium, is far more learnable than programming languages like Python, which are (by default) by and large aimed at already programming-literate folks. We designed Boxer explicitly to be significantly easier to start programming in than Logo, the benchmark I "grew up" with, just entering the technology and education field. I think we succeeded admirably (diSessa, 1991). In fact, early results working with teachers and students were so positive that we just gave up on a project to prove learnability, in favor of simply starting to teach with it.[3]

A second technological dimension is expressiveness. This relates directly to the re-mediation idea in my five powerful ideas exposition. How does the medium make it easier to express and explore ideas from a wide range of arenas, such as much or most of traditional school mathematics or science? This is transparently important for the goal of computational literacy, but it is simply not a visible goal of "garden variety" programming languages. Scratch most definitely was designed with learnability in mind, but my own judgement is that it has a pretty hard limit in terms of expressiveness.[4] I will just note, again, that expressiveness (for non-technical users) has very little, if any, presence in the design of most programming languages, and considering the issue brings us into the cognitive worlds of students in relation to an analysis of subject matter. How well does any computational system do at expressing the idea of "force" or "prime number"? Expressiveness is also simply not the same thing as comprehensibility of computational systems. So we see diverse and very different causalities coming together in designing and/or evaluating technology for a computational literacy.

A very different causality (very different science) undergirds the social/cultural dimension of literacy and its development. How do we understand the various subcommunities that might adopt a new literacy, their characteristic goals and capacity to change from existing practices? In order to consider the cultural dimension, one would not even be tempted to consult the same people as for the technical dimensions. Academia is famously "siloed," definitively separated, especially as it relates to computational literacies. The bridging that is necessary in considering new-literacy development needs to happen elsewhere than inside existing academic disciplines.

I'll have to leave the topic of multi-causal science now. Mostly, I just want readers to have a rough sense of what is involved, and then to be in awe of how complicated the issues are. We have miles to go to get the relevant sciences to speak clearly, in concert with each other, to optimally design for and evaluate new-literacy initiatives.[5]

---

[3] There were some technical and careful published results, even from outside the Boxer community. Results of a study done by scientists at IBM Germany about constructing complex data objects were so stunning that there was notable public pushback, which I think was safely be dismissed as a kind of "sour grapes." Some of our own results with sixth graders involved topics like scoping, which is seldom even discussed with novice programmers.

[4] Obviously, like many other elements of my argument, here, this deserves elaboration, which I won't have space to explore here.

[5] I think in the short term—or possibly even in the long term—we will need *people with good intuitions* about many/all of the diverse collection of causalities involved in new literacies, while the different relevant sciences get their acts together to address literacies and their development adequately. You might think that the theory of literacy would be a good place to look for a unified treatment of the multiple causalities involved. Sadly, this is not in the offing. Looking through the history of literacy theory, early and mostly purely cognitive approaches have said virtually nothing about the sociocultural side of things. Newer, sociocultural approaches seem to have completely rejected any cognitive insights, such as re-representation. And, the design of computational media, meeting the demands of new literacies, is such an exotic enterprise that I would be daunted to try to build a decent reading list.

What I'd like to do now is to discuss how computational literacy will or won't look like textual literacy, which is the prototype reference model. As I mentioned, while it is speculative, this discussion is also grounded in the many experiences we've had using and teaching with a computational medium. I'm going to proceed by examining the repertoire of "objects of literacy" that are parts of textual literacy and/or will be part of computational literacies.

"Objects of Literacy": Size and Patterns of Re-use

I want first to consider the size of *objects of literacy*, and, more generally, the distribution of them across the spectrum of sizes. Maybe the prototype for literacy—and arguably the most important class of objects involved with textual literacy—is books. These are pretty large, as scales of literacy objects go. But, when we think of "great literature," especially that which has been most influential to civilization, books most definitely come to mind.

Thinking across a span of sizes, "aphorisms" and "quotes" occupy the smaller end of the textual literacy objects spectrum. These are broadly apparent but not nearly as important as book-level literacy objects.

Will there be book-sized objects of literacy in computational literacies? In fact, the image of a book, augmented internally by computational objects—such as simulations and microworlds (learning/exploring environments targeted at particular topics) was one of the guiding images for the creation of Boxer.[6] And it's an image that I have found resonates with many people. But, here's a cautionary tale about where we are in the development of computational literacy: I am now writing a physics textbook that would absolutely work marvelously as a Boxer book. However, I feel I simply can't risk all the ideas I aim to put into this book just falling by the wayside of technology change. Textual books have lasted centuries. Computational books, right now, would have a very uncertain future based just on the technology available to support them. So, assuming computational literacies develop, "books" may, ironically (in view of their importance) come late in the game.

Difficulty in developing long timescale, book-sized contributions is one reason that smaller-sized objects of literacy may predominate, at least early in the development trajectory. But, I think the prevalence of smaller-sized objects gets a more general and important boost in computational literacies. Quotes and aphorisms have their place, but little tools and computational objects of various sorts have proven far more durable in their use than corresponding textual objects. I think the key is that those small things *do* things and can be used in a lot of different environments with other computational (and textual) objects. Quotes and aphorisms, to take the same "small/textual objects" examples, tend to be very specific, and are not malleable in the way that computational objects can be. Those who saw my lecture should have noticed a bunch of standard little tools—like buttons, "clickers" (to turn things on and off), sliders, even vectors—that can be cut and pasted into (and easily computationally integrated with) any "new" microworld or work environment. Color pallets provide a good "transfer" example. They were developed for an image-processing toolset but could then be just cut and pasted into the Chartworld environment to allow students to pursue aesthetic aspects of number patterns. See also diSessa (1997) for more examples and an elaboration of the "flexible, re-usable computational objects" idea. You can write me to get a copy of that paper.

I make here just one more observation about the size of (computational) literacy objects. We have found that simple versions of tools and other objects are inordinately powerful. First,

[6] Boxer's core organization is a hypertext processor—text augmented by hierarchical structure (boxes inside boxes) and links across them, which are called "ports" in Boxer.

they can do an important job quite directly in many contexts, while being simple enough to be easily understood. And, since they are easily inspectable and modifiable (prime goals of Boxer), they can be simply adapted in small doses to work better in different use-cases. Contrasts are both (1) "*apps*" (e.g., iPhone apps) that are almost completely closed to modification or for re-use in different-than-originally-intended contexts, and (2) large *applications* such as Microsoft Word, which, in combining resources for every imaginable use, become overly complex and difficult to use. In Word, I always have trouble finding certain resources I need for various purposes, even though I know they are available—somewhere—in the application.

To summarize, computational objects of literacy will appear at every scale, similar to conventional literacy. However, I predict a strong "tilt" toward smaller scales, because of the ease of sensible modification and re-use in different contexts, in conjunction with other small (or larger!) computational objects. The long path to a true literacy is implicated in the cautionary tale that some of the most powerful objects of literacy (book-sized objects) are hampered during this, the early phases of literacy development, by comprehensible but potentially catastrophic "accidents" (such as a shifting infrastructure for computational media).

A Manifesto?

Brent made a point of asking me whether the paper he read was a "manifesto." Or, possibly, he was asking whether the very idea of "computational literacy" *constitutes* the core of a manifesto. Without pursuing his actual intentions, I'd like to provide some clarity.

The first two "hits" in a Google search resulted in the following: A manifesto is: "… a written statement of a person or group's beliefs, aims, and policies, especially their political beliefs"; "… a call-to-action that works to ignite the masses." A couple of edits adapt the definition to the present case. First, computational literacies are not political in the common meaning of that term. At the top level, I don't perceive a difference between the political right or left with regard to whether people should learn to read and write. (Of course, the *content* of what's written about is a different matter, but not obviously related to learning the medium.) "Policies" might be a little questionable in the context of literacies, too, given its normal association with politics, which I've already jettisoned.

The second edit is about "igniting *the masses*." A more sensible adaptation, in this case, is "*everyone*, especially those interested and capable of moving the agenda forward."

These aside, I'm happy to accept the claim that I am often advocating for computational literacies and urging action. In that sense, some of my writings may be properly regarded as manifestos.

On the other hand, I think it is very important to keep clear on the fact that "manifesto" is a rhetorical form that has very little to do with the nature of the thing that is being advocated. I think it warrants extra care to keep the default political context for "manifesto" at bay. The nature of the thing I am advocating is quite distinct from that for which manifesto is usually invoked. I would say, at least, that the object of my advocacy is:

Scientifically accountable: Even though the sciences of the various causalities that are involved with literacy are not sufficiently developed or integrated to be definitive, I most definitely have scientific accountability in my sights. My community and I have written papers about: (1) how to understand and evaluate comprehensibility of computational system (e.g., diSessa, 1991), (2) how computational representations actually work, in the moment-by-moment thinking of students, to develop scientific concepts (e.g., Parnafes, 2007), (3) how these representations can evoke relevant student intuitive knowledge, to support their learning productively. In all these pursuits, I am open to the usual scientific scrutiny, evaluation, and dispute.

Practically accountable: In the end, this whole enterprise, in my estimation, comes down to "can we substantially improve student learning of science and mathematics in the mode of computationally literate practices?" Many of my own (and those of my extended research group) published results are various versions of "look how much earlier we can teach X than is presently true," or "look how we can teach important things that have never been very teachable with non-computational media" (see diSessa, 2008), or "look how we can make the learning process much more enjoyable and engaging for students, without vitiating the ultimate goal of having them come to a deep understanding of science and mathematics."

To counter the (hidden but perhaps insidious) influence of the standard political use-context of "manifesto," and also to be truer to the nature of computational literacies as an object of advocacy, I might prefer the term "programm" (from the German, or as sometimes rendered "program" or "programme"). My personal favorite example from mathematics is Klein's Erlanger Programm. In his monumental 1872 paper, Klein advocated a large and long-term agenda of integrating geometry and group theory. His paper was regarded by some as rather "philosophical" rather than typically mathematical. But the effect of that paper was, over the long haul, exactly what Klein was advocating.

> In 1872, the E.P. [Erlanger Programm] was 20 years ahead of its time; it would take at least that long for the new perspectives of Klein and Lie to gain general acceptance. However, their ideas would take deep root; in fifty years it would become commonplace [to fully embrace Klein's programme]. (Birkhoff & Bennett, 1988, p. 152). [With minor edits and clarifications of the original quote]

At 20 years, computational literacy might be a little behind the standard that Klein set. But, given 50 years, perhaps advocates of computational literacy, as a programme, can catch up. That is well within the time-scale of what I have taken to be a sensible expectation for really developing a broad computational literacy: 100 years (given precedent for historical development of literacies, specifically both mass literacy with text, and with algebra/calculus).

Avoiding getting "Winged"

Brent asked the basic question of how we can look past the noise of contemporary buzzwords and transient obsessions in education to get to the crux of profound, long-term progress. My somewhat facile answer is, of course, to attend to the idea of computational literacy. This larger frame, valuable especially for literacy skeptics, allows us to be critical and look for robust science at the base of any possible direction, and to look to people who have a commitment toward actually changing teaching and learning with technology, people who present us with innovations and good documentation of "what happened, how that happened, and in what way is it better."

Jeannette Wing's "computational thinking" is my constant reference for "currently popular things to avoid." So, I propose to run through a quick-and-dirty evaluation of Wing's conception of computational thinking, in direct comparison to each of the five powerful ideas that I proposed as worthy directions in my talk. I add sample references to the work of my community in cases that previous citations in this article have not touched.

1. *The very notion of computational media and new (computational) literacy*: There is no even minimal reflection of anything that is identifiably in this direction in Wing's writings. There is no mention, for example, of the critical task of developing a *literature*

for new media. Indeed, there is no mention of learning any particular science or mathematics, per se. Instead, quite clearly, Wing invoked the questionable—and I think discredited—idea that special disciplines (like computer science, or, historically and famously, Latin or mathematics) uniquely develop general powers, such as abstraction, that make performance in every disciplinary context better.

2. *Re-mediation*: Computational literacy, in my view, inherently draws on the idea that computational representations can be more apt than many others for particular arenas of science and mathematics. It cannot be universally optimal; every representational system has strengths and weaknesses or blind spots. But my claim is that computation is both easier to learn and far more applicable to a diverse range of scientific and mathematical things-to-learn than representational systems such as graphs or even algebra/calculus. A lot of the enterprise is to find the strengths of computational representations, capitalize on them, and to find other ways to improve instruction when computation alone is insufficient, or, indeed, not directly helpful at all. Wing's sole bridge to particular subject matter is via skills that are simply not particular to any subject. They are, by claim, domain neutral.
3. *Engagement and Activity Structures*: Wing has explicitly advocated making computer science instruction more fun and engaging. But she has nothing to say about making specific science or mathematics similarly more engaging. As far as I am aware, there is precious little literature on programming, per se, making the learning of anything else more engaging. In our work, we have not only sought engagement as a goal, but researched the ways in which some topics or engagement strategies work well…or poorly. (Sherin et al., 2005).
4. *Open Toolsets*: Wing does not treat in any way how programming, as it exists, is limited in advancing educational goals. Open toolsets, in contrast, is a partial architecture for learning objects that has special advantages over "just programming." These include adapting quickly to particular needs (such as different subject matter), and the ability to easily co-opt resources from one educational application into another.
5. *The LaDDER model*: As far as I have read, Wing never explicitly mentions culture or social elements of developing the competencies she aims for. The whole sociocultural layer of causality is, thus, missing. The LaDDER model is, in contrast, a social architecture for overcoming cultural divides and suboptimal concerted work that we observed in real-world empirical data about educational software development. It thus represents the core, larger observation that a new literacy is, at its heart, an accomplishment of a culture or society. It is not simply a thing in any person's head, conveyed by programming. diSessa, Azevedo, and Parnafes (2004) provides a slightly extended and situated discussion of the LaDDER model and the problems it addresses.

Beyond this "five-ideas" list, I am unaware of any scientific literature showing that Wing's program actually works to make people "smarter" or passes any test of scientific legitimacy.

Forward!

To close on a positive note, I repeat my response to Brent when he asked for guidance on which of the very many paths, implicit or explicit, in my five powerful ideas talk should be pursued urgently, now. For myself, I have chosen to explore new things to teach (radical changes in the very curriculum of schooling), and simultaneous improvement in student engagement. These are the most exciting and important possibilities I see, and I have a special affinity (and maybe I've developed not-so-common skills) for pursuing them. But there's such a broad front that is open to us that I feel no particular need to proselytize for my own interests and preferred directions. We'll all benefit from diverse engagements in the larger program.

References

Birkhoff, G. & Bennett, M. K. (1988). Felix Klein and his Erlanger Program. In W. Aspray & P. Kitcher (Eds.), *History and philosophy of modern mathematics* (pp. 145-176). Minneapolis: University of Minnesota Press.

diSessa, A. A. (1991). Local sciences: Viewing the design of human-computer systems as cognitive science. In J. M. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface* (pp. 162-202). NY: Cambridge University Press.

diSessa, A. A. (1997). Open toolsets: New ends and new means in learning mathematics and science with computers. In E. Pehkonen (Ed.), *Proceedings of the 21st Conference of the International Group for the Psychology of Mathematics Education*, Vol. 1 (pp. 47-62). Lahti, Finland.

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

diSessa, A. A. (2008). Can students re-invent fundamental scientific principles?: Evaluating the promise of new-media literacies. In T. Willoughby, & E. Wood (Eds.), Children's learning in a digital world (pp. 218-248). Oxford, UK: Blackwell Publishing.

diSessa, A. A., Azevedo, F., & Parnafes, O. (2004). Issues in component computing: A synthetic review. *Interactive Learning Environments*, *12*(1-2), 109-159.

Parnafes, O. (2007). What does "fast" mean? Understanding the physical world through computational representations. *Journal of the Learning Sciences*, *16*(3), 415–450.

Sherin, B., Azevedo, F., & diSessa, A. (2005). Exploration zones: A framework for describing the emergent structure of learning activities. In Rosebery, A., Warren, B., Nemirovsky, R., & Solomon, J. (Eds.). *Everyday matters in science and mathematics* (pp. 329-366). Wahwah, NJ: Lawrence Erlbaum Associates.