**Upshot:** Pavel Boytchev’s article calls attention to the fruitful dialectic between building things and taking them apart: No successful construction without deconstruction. Of course by using the word “deconstruction” he is also implicitly invoking the critical-theory sense of the term, inviting us to deconstruct Constructionism. I found the article fascinating on both levels.

1. This is a very provocative target article. Its central thesis, that deconstruction is essential to successful construction, is an idea that, once stated, is obvious, but it is far out of the mainstream of constructionist vocabulary. I think, though, that the *practice* of deconstruction may be more prevalent than its analysis.

2. For one thing, it is a commonplace observation that you learn how to build things by looking at, and looking inside, already-built examples. Sometimes this involves physical deconstruction, as in Pavel Boytchev’s example of the child deliberately breaking a toy to see how it works. But I am thinking also of the “Look inside” button on Scratch project pages. The visitor’s first view of someone else’s Scratch project focuses on its behavior and purpose. But by clicking “Look inside,” the visitor can examine the Scratch program that makes the project function, and can even modify or “remix” it.

3. Secondly, people try to build learning environments based on a small number of simple parts. The canonical example is Lego, which has been much more successful than the Erector sets of my own childhood, in which the main components were metal bars of many different lengths that could be bolted together, along with specialized connectors such as angle brackets, in part because of the wide range of projects that can be built out of (many copies of) a single part, the Lego brick. Such environments come pre-deconstructed. The same could be said of the effort in the design of Scratch to minimize block shapes and to minimize the number of primitive blocks altogether.

4. Thirdly, the emphasis on physical stuff in the Maker movement makes the component parts much more readily visible than in either computer software or even more abstract mathematics.

5. I would be interested in a discussion of how the idea of deconstruction fits in with the (I think) overlapping idea of demystification. One way to demystify a black box is to see how it is built. But there are also demystification strategies that, for example, try to shed light on people’s motivation, or on sociological ideas such as “institution.” Is the purpose of deconstruction demystification, or something else?
6. I think some of the points in the paper are made in too much of a hurry. Here are two examples:

7. Debugging as deconstruction (§30): I get the overall point, I think; in order to debug you have to think analytically about the pieces of a program. And I get that current debugging technology is weak. But I do not think I quite got how a deconstructionist perspective can lead to better debugging tools. This could be a research project in itself, and the one paragraph here leaves me dissatisfied.

8. Technology and education (§43): There is a claim here that very rapid change in technology makes it hard for education to keep up. I see how that would be true for education about technology – sort of the point of the Noam Chomsky quotation in §43. And I can see how education might not be making the best possible use of technology, although, for example, schools have been quick to jump on the tablet bandwagon, but as far as I know they have not yet found any purpose for which tablets are better than laptops. (I do get that they are cheaper, although now you can get “netbooks” instead.) But children are not so very different in how they construct knowledge, are they? If that is the intended claim, it needs way more discussion. Otherwise the point about exponential change is a little glib.

9. By contrast, the discussion of the virtual classroom project is detailed, vivid, and very convincing. Figures 8 and 9 are a wonderful encapsulation of the central point. I would have been glad if the last few pages of the paper were more a laying out of a specific program for future research, rather than a collection of brief mentions of largely unsupported ideas.

10. I do wonder why students of the virtual classroom were required to deconstruct not only the physical model being simulated but also the simulator as a piece of software. Why, for example, make a point of not documenting which keystroke does what in the user interface? Doesn’t that deconstruction task just distract from the deconstruction of the classroom environment?

11. There is another paper to be written about the deconstruction of education. Does that mean something more than the critique of education that starts (maybe) with Jean Jacques Rousseau and goes through John Dewey to Paulo Freire and Ivan Illich? Is there some way in which computers change this critique? What are the component parts of education? Learner, teacher, and content? Maybe plus the learning community? Or does Boytchev mean something more detailed than that? When I visit a school, I generally find myself thinking first not about technology, but about the social relations in the classroom – mostly the one between the teacher and the students, but also relations among the students.

The author

Brian Harvey is Teaching Professor Emeritus in computer science at the University of California, Berkeley. He has taught every age from eight to adult, teaching programming to the kids and the teaching of programming to the adults. He is the
author of the three-volume *Computer Science Logo Style* books for teenagers. His recent efforts have been co-developing the Snap! visual programming language (with Jens Mönig) and the “Beauty and Joy of Computing” curriculum (with Daniel Garcia).

Received: **to be completed by the editors**
Accepted: **to be completed by the editors**