DavidCrusoe4

(Hopefully, this makes sense! I'm trying to mesh lots of stuff together here, so feedback is gladly accepted!) -- dave

---

In considering this week's readings, I find myself my mind exploring differences between the concrete and the abstract, the physical and computational. For example, is it possible for a physical object to transcend it's "simple nature" to become something more? Or, on the other hand, for a computational object, in the hands of an unimaginative learning session, to be relagated to a very physical level? Furthermore, is it the compuatational object that spawns abstract thinking, the physical object that spawns concrete thinking, or vise-versa? Or, in the hands of an expert, could an object such as a hammer become a "toy to think with," and, at the same time, an object to concretize the abstract, as well as an object to abstract the concrete? Does it help, then, to be an adept "bricoleur," able to piece together both "hard" and soft" understanding in order to create a picture of the whole?

For example, an expert might learn patterns of building by using a hammer, and might utilize those patterns to explore a novel structural problem. While actually building the solution, the expert might recognize a new design problem to tackle, and so the cycle repeats. Eventually, a system of thinking emerges by which old patterns help to formalize new patterns: transfer of a rules-set?

To add many layers of complexity is to arrive at a computing machine, which essentially hammer and saws to a very specific set of rules, ie., an explicit system of thinking and "programming." So, for example, while I can't imagine all the processes inside my computer (and all the transistors that are firing) to create the text that I'm writing now, I could theoretically program a layer upon these rules to create the program to do so. Eventually, I understand how the machine accepts rules, and so I'm able to innovate.

In both instances, the concrete serves to reinforce the abstract. As a recent NSF report recognizes, "Cognitive research has converged on the conclusion that transfer is better if people have learned inititally in a way that fosters deep, abstract understanding of central principles of a field..." (Transfer of Learning, NSF 2002) Furthermore, while Turkle and Papert (1990) suggest that [the computer] can "make the abstract concrete," I argue that in the hands of an expert learner, the proper tool (as with the hammer case above) can do so. Innovation in either case, then, relies upon expertise or accident.

Depending on my purpose, these tools either serve me as "Black Boxes" - tools for input/output or "computational objects - "toys to think with." Of course, depending on a tool, the sense is slightly different; a hammer can't compute (or, isn't programmable (yet)), a computer can't hammer. But in considering the thinking that I've applied to each instrument, and depending on the purpose, I've used each both way.

(more coming as I formalize my thoughts!)

-- 11 Oct 2004