

Learning Isn't a Push Model

by Kathy Sierra

Back in my AI days (when I used to have a clue, or thought I did anyway), the book *Scripts, Plans, Goals, and Understanding* was required reading for some of us. And I've been a fan of Roger Schank ever since. Of all the work that has influenced the direction of our learning principles, his has had the greatest impact. We were thrilled to see his work on intelligence move toward developing better theories (and tools for) learning.

We get letters from people who want to know more about the metacognitive topics we cover in the intro to the *Head First* books, and I'd suggest that anyone interested in learning theory put his e-Learning book high on your list. Maybe even at the top.

Some consider him an acquired taste, and he has a lot of detractors. He's one of the more outspoken critics of the education system in the US and slams just about everything from secondary schools to colleges to corporate training. A typical quote of his from the book, discussing corporate training:

"So what's wrong with training? Everything that's wrong with training can be stated in four words: *it's just like school*. The educational model in school does not work. That fact, however, hasn't deterred business from adopting this model, which is based on the belief that people learn through listening. Memorize the teacher's words; memorize the training book's policies and procedures. It's at this point in my public talks that audience members rise up in protest."

And one of my favorites:

"First and foremost: When learning isn't engaging, it's not learning. The movies, for all their faults, usually get this idea right. In the film *Dead Poet's Society*, Robin Williams plays a teacher who jumps on top of desks, makes the class laugh, tells great stories, and gets the class involved in what he's teaching. The educational establishment at the school hates the way Williams teaches, based on the premise that if students are having fun, something must be wrong. Listening to lectures and memorizing countless facts and figures aren't engaging activities for most people. Minds wander; real goals take over."

Another book that has some good research data is this one by long-time learning guru Ruth Clark.

Yes, both of these books happen to be focused on e-learning, but the principles apply whether you're doing classroom training, learning books, online training, or developing just-in-time performance support materials.

But regardless of differences among learning theories, one thing virtually *everyone* agrees on is this: **Knowledge cannot be pushed into someone's head while they sit passively reading or listening. Knowledge is a co-creation... the learner must construct the new knowledge in his**

own head. And usually (or some say ALWAYS), the new knowledge must be mapped into something that's already IN the learner's head.

This notion of knowledge-as-co-creation is crucial for us. Which is one of the reasons we were horrified at the thought of creating learning books. Because for the most part, *reading* is just like *listening*. Worse, reading a fairly dry text book is like listening to a dry lecture-- pretty much THE weakest form of learning. So trying to make a book into a learning experience flies in the face of everything we believe in about learning (our backgrounds are in computer science, game development, entertainment, teaching, and loooooong stints in artificial intelligence including the field of intelligent tutoring systems (AI meets CBT)).

So our mission was, given the constraints of a book format, and knowing that learning is far less likely to happen if the learner is just... *reading*, what can we do to help get them involved and start *flexing their brain cells*? So we tried a bunch of different things, figuring that the more we can throw at it, the more likely it is that *something* will work at least *some* of the time, for the people who try to learn from the book. From the feedback we've gotten now, 18 months in, we know that some of what we did to help make this happen is working, and some of the things just bombed. And some of the things we didn't plan--that are there simply as artifacts of trying to apply some *other* principle, turned out to be a key component to getting the learner involved.

Getting the learner to co-create knowledge isn't a simple task; lots of pieces have to come together. For example, if we provide the absolute best thought-provoking exercises, but can't get the reader to stay awake long enough to get to them, we lose. If we provide ways in which the reader can get involved and really build some brain cells, but the challenge level is simply too high (or just as bad, too *low*), we lose. If the material simply isn't stimulating enough to hook the reader in, and he won't stay with us start to finish, we lose (since we're not a reference book). In other words, our whole premise and promise to the user--that they'll learn more quickly, more deeply, and with less pain--depends on them *really staying with it* and doing the work. And we're painfully aware that if we don't deliver on that promise, they'll have no reason to ever buy another book in this format. We're in this for the long haul, so we're deeply dedicated to really making this work for people who want to learn.

We looked at the whole system in which an environment for learning occurs, and that's why we drew on so many domains to help us. Learning theory says the learner must be motivated, but says almost nothing about how to *get* people motivated. Learning theory says the learner must be engaged and involved, but says almost nothing about how to really make that happen. On the other hand, Hollywood, Madison Avenue, and good sales people have something to teach us about providing motivation. They understand seduction and communicating *personally meaningful benefits*. (More on that topic in other posts.) Game designs have something to teach us about that as well. After all, kids and adults alike will spend hours and hours and hours immersed in thinking/planning/strategizing in the course of playing a video game.

We wondered, can we try to turn a technical book into something that will make people *want* to stay with it and keep turning the pages? And even if we *can*, will they be motivated to

actually DO the exercises? What if they *don't* do the exercises? Can we provide OTHER ways to try to make learning happen? To get the user to think at a higher level... to process the new content in such a way that he constructs new knowledge in his head, rather than just passively reading?

Our answer was, "We don't know if we *can*, but yes, we can certainly try, and here's how..."
Among the things we use to try to get people flexing brain cells are:

1) We use cliff-hangers, where the learner is drawn into the scenario only to be left hanging without the full answer, to help spur their curiosity into speculating on the solution or result.

2) We use debates/arguments/discussions between two characters (which could be people or even anthropomorphized parts of the system like the compiler vs. the virtual machine) where there isn't always a clear winner. Both sides might make compelling, convincing cases for their personal view, and this kind of forces the learner's brain into evaluating (one of those higher-level thinking tasks on Bloom's taxonomy), weighing the merits of each side, and drawing his own conclusions. Sometimes we have a definite side, but looking at the same scenario from more than one perspective is in and of itself a way to help inspire deeper thought processing of the concepts.

3) Knowing that most people claim to skip the lab exercises in a book that say, "Now go to your computer and type this in...", we have 40-50 in-book (workbook style) exercises you do with a pencil, right inside the pages of the book. We want readers/learners to have NO excuse for not doing the exercise when they're using the book the way they tell us they do--on the train, bus, plane, in the park, wherever they have a spare moment at lunch, etc. In other words, when they're not necessarily within easy reach of doing the real thing they're learning.

4) We make those exercises use a wide range of brain capabilities--so there are right-brained pattern-matching exercises, left-brained code troubleshooting and logic puzzles, draw this picture, answer this question, write this code, make this decision, etc.

5) We ask questions and provide exercises sometimes for things that we haven't fully explained, so that the learner must *apply* what he's already learned and extrapolate to figure out something he hasn't actually seen yet.

6) We provide "garden path" scenarios, where the learner is led down a road that looks so right, but turns out to be SO wrong. This is based on the theory (Roger Schank has a lot to say about this as well) that we learn a lot more from mistakes and surprises than from things that work as expected. Think about it... what are the things you're most likely to remember when you're working? When things go just as you expect, just as you were told, there's nothing memorable. But when you're humming along and suddenly the thing you expect fails, and you get just the opposite... you get that WTF?? feeling. And **that** is what you remember. So we try to provide at least a few of those visceral, "I won't make THAT mistake again!" experiences when they happen. (And thanks to the wonderful Java APIs, those doesn't-work-the-way-you'd-think counterintuitive scenarios are plentiful in some of our books:)

Of course it's a little tricky using these techniques. It makes our books suck as reference books, of course, but we're 100% clear that our books are learning experiences, not references. Because you might flip back to a page and actually find yourself *reading something that's just wrong*. So we have to use a lot of other things in our books to try to get people to read the whole topic, at least the first time through, in order to get the whole "Here I am doing this stuff and BAM! It blew up..." experience of the story. And there will always be some people who hate the approach precisely because it DOES include these tricks. They feel cheated that we deliberately led them down a garden path when we could have just *told them how it really works to begin with*. And while they have a valid point, that "telling not showing" approach (considered a really bad thing to do among novelists and filmmakers) is the weakest form of learning.

You might hate the approach, but it's closer to the messy, confusing, oh SHIT experience of what happens when you do these things in the real world, and it's guaranteed to be more memorable. If you can stand getting through it. That's part of why we have to do a lot of other things to try to make that "getting through it" more tolerable, or even interesting. (That's where the game design theory comes in; more on that later, but here's a hint: the flow state psychologists call optimal experience that game designers know as "make them want to get to the next level by getting the challenge/skill/seduction blend right.")

Source: http://headrush.typepad.com/creating_passionate_users/

Reprinted with permission: K. Sierra 2008