



My Own Kind of UML: Innovative Technology Transfer With the Universal Motion Language

I've been thinking a lot about why it is so hard to get software right, and I think I may have hit on something.

I have noticed when visiting with relatives who live in the country that they are mechanically astute. They can fix or jury-rig almost anything. They work on problems at least as complicated as, say, running an automated payroll system for hourly employees. Have *you* ever tried to replace the boot on a compensating velocity gear on a tractor, or unspool an old fishing line reel and not lose that eight-pound bass? See what I mean!

What is of particular interest to me is their multi-channel knowledge transfer methodology. When describing a technique or process, conversations are punctuated and animated with hand motions and body postures. I get tired just watching one explain how to tighten a strand of barbed wire with nothing but a broomstick and some seagrass string, or how to pluck a chicken for supper on the kitchen table when it is storming outside.

You can look all around and see different groups with their own lexicon of hand and body motions: baseball coaches instructing a runner to steal, steelworkers guiding crane operators as they put I-beams into place, or a driver expressing disdain—with merely one finger—when another driver cuts them off.

Software developers are no less creative and resourceful, but are clearly missing the hand motions and body postures. I recommend computer science centers of excellence immediately pair up with psychologists in academia and begin developing these visual clues forthwith.

As a departmental process expert, it is my job to introduce and explain the latest processes and forms required for all projects. In my opinion, visuals are needed.

I have worked up a few on a trial basis, such as inserting a new item in the middle of a doubly linked list. I find the hand motions needed to help explain how to create a new relational table to be much less complicated than the doubly linked list—though initial trials with my colleagues on the job have so far proved unproductive.

I have also attempted the utilization of several hand signs, such as illuminating the walk-through process using two fingers walking across the desk. Some of the trainees responded with their own lower energy-state hand sign using less motion ... and fewer fingers. Hand and body motions are often more efficient (requiring less mental effort) than words in expressing concrete ideas and procedures. It is easier to describe a spiral staircase with your hands than it is with words.

In my humble opinion, these less than enthusiastic reactions are a case of new technology being rejected by seasoned veterans who fail to see the value in something they didn't invent themselves. It is, as they say, "not the way we do things around here."

I wanted to call this new technology UML, Universal Motion Language, but I am told that acronym has already been taken by some upstart language those smarty college

new-hires keep wanting us to use¹. I also toyed with calling it Associative Symbolic Signing, but my wife nixed that name. She suggested Symbolic Transfer Using Polite Indicative Directions.

The need for this new technology is accelerating faster than Moore's Law² as new ideas for improving software development productivity are popping up, I suspect, at roughly two new ideas every 18 months. Concomitant with the new ideas are their techniques, processes, models, books, and conferences, thus driving up the need for better knowledge transfer techniques exponentially.

We are almost to the point where we need both an XML open gesture tag and close gesture tag to identify the ever-growing list of standard gestures and distinguish them from other more prosaic hand motions and postures, such as combing one's hair, scratching one's posterior, or testing the pH of one's azalea bed.

Adapting this technology for supply chain automation activities should be a great place to start since the hand signals for driving a truck, pulling inventory, and shelf restocking are fairly intuitive, while request-for-price and backorder transactions would be more problematic.

But if we make this technology "freeware" and also give away rather than sell the documentation, then the open source and freeware communities can make short work of those type problems. After all, no problem can withstand 10,000 hands.

—Carl Wayne Hardeman
cwhardeman@yahoo.com

Notes

1. They also suggested adopting a technology they call PowerPoint. We told them that is not the way we do things around here.
2. Moore's Law, which dates back to 1965, states that the number of transistors on a chip will double every two years. See <http://en.wikipedia.org/wiki/Moore's_law> to learn more.

Can You BACKTALK?

Here is your chance to make your point without your boss censoring your writing. In addition to accepting articles that relate to software engineering for publication in CROSSTALK, we also accept articles for the BACKTALK column. These articles should provide a concise, clever, humorous, and insightful perspective on the software engineering profession or industry or a portion of it. Your BACKTALK article should be entertaining and clever or original in concept, design, or delivery, and should not exceed 750 words.

For more information on how to submit your BACKTALK article, go to <www.stsc.hill.af.mil>.