

Membership: Member Profiles



Dr. Eddie Pettis

IEEE-HKN Governor at large,
Platform System Engineer, Google

Dr. Eddie Pettis has served IEEE-HKN as Governor at Large since 2009. Dr. Pettis was inducted into the Gamma Phi Chapter of Eta Kappa Nu at the University of Arkansas in 2001. He served in multiple offices for Beta Chapter during his Ph.D. program at Purdue University from 2003 to 2008. He has received numerous academic awards, including the University of Arkansas' Chancellor's Scholarship, the Branigan First-Ranked Senior Scholarship, and the Bilsland Dissertation Fellowship, and was named a finalist for the Chorafas Top Doctoral Student Award for the Purdue University College of Engineering. Since his graduation, Dr. Pettis has designed embedded systems and power management software for Google's data centers. He lives in San Jose, California with his wife, Charity (also an IEEE-HKN member, Gamma Phi 2001).

Why did you choose to study the engineering field?

When I showed up at orientation, I didn't know what I wanted to do. However, I was in love with my guitar. Deep down, I think I really wanted to major in music, but with two band directors in the family, I knew how that road ended. I enjoyed mathematics, so I decided to mate the two. I chose to study electrical engineering because I wanted to build my own guitar amplifiers and effects pedals. My interests eventually shifted towards digital electronics and embedded systems, leading me towards graduate study in power management. In retrospect, I guess it was fortunate that I didn't major in music.

What do you love about it?

Anything involving electromagnetics. I got an "A" in the course, but I have never felt comfortable with it.

Who do you admire?

Steven Levitt and Bill James.

Levitt is the economist behind "Freakonomics." His work is absolutely fascinating and easily accessible for the analytical reader. He takes a rigorously quantitative approach, rather than accept hollow conventional wisdom. His work is also quite controversial, and I have a soft spot for people who challenge the status quo.

Bill James revolutionized baseball from his Kansas home by bringing quantitative analysis to the game. James built his own reputation by challenging assumptions and backing them up with mountains of data in "Baseball Abstract." His work completely changed the way players were drafted and in-game strategies.

How has the engineering field changed since you've started?

Modern computing has become power-constrained, and it is changing everything.

On the desktop and server side, we've migrated from a world where most problems were solvable by throwing more hardware at the problem to a world where concurrent programming on many lower power, lower performance cores is required to achieve high performance. There remains no easy way to handle this. Some say that we can switch to functional languages, but this ignores all the existing procedural code bases. Procedural code bases lack the support for automatic parallelization, resulting in error-prone modifications.

On the mobile side, battery lifetime is a constant problem. This is often resolved through a variety of hardware solutions: CPU sleep states, increasingly low-voltage cores, fine-grained power domains, and higher energy density batteries. Although hardware will always remain a viable solution, software is increasingly important, from the operating system to the application. Operating systems support tickless kernels to enable deeper sleep states. System daemons monitor CPU utilization to scale voltage and frequency intelligently. Applications offload expensive computations to servers when the computation energy exceeds the transmission energy.

None of this was being taught in school when I was an undergraduate. The problems were only being taught at the graduate level when I was in graduate school. We're becoming a mobile society, and we're going to have to start stressing that to undergraduates.

What direction do you think that the engineering field is headed in the next 10 years?

Engineering as a whole? That's an intractable question!

Going back to the previous question, I think that simplifying concurrency is the problem of our generation. It's tempting to say that new programming languages solve the problem, but rewriting everything is impractical. It's not that we can't rewrite everything; we simply

won't rewrite everything. There's billions of lines of C and C++ out there that simply aren't going away. Somebody is going to find a way to parallelize C and C++ automatically, and they're going to become filthy rich.

What's the most important thing you've learned in the field?

To paraphrase Bill Coughran, former VP at Google, "Technical problems are easy. People are hard."

We rarely fail because we lack technical expertise. We fail because we duplicate effort. We fail because the hardware developer didn't know that the software developer needed a faster sampling rate. We fail because I think the problem is important and you don't. We fail because I blocked your working solution when it made my job harder. We fail because managerial bureaucracy poisoned team morale. We fail when the team continues to optimize instead of shipping a "good enough" product. People problems happen everywhere, and they destroy even the most technically savvy organizations.

What advice would you give to recent graduates entering the field?

1. Be excellent to each other. I work with lots of talented people, but nobody wants to work with negative people.
2. Fix the problem, not the blame. Your boss's boss doesn't care who screwed up, only that the job gets done on time.
3. Accept credit when it's due, and acknowledge others in their successes, especially when they contributed to your success.

If you weren't in the engineering field, what would you be doing?

Economics and college football analytics. Both deal with a complex collection of variables in a sea of noise. Extracting useful information from the system is incredibly difficult. I enjoy the challenge so much that I write algorithms to pick college football games as a hobby with one of my friends from work (<http://tempo-free-gridiron.com>).

Finish this sentence. "If I had more time, I would..."

... learn the technique necessary to play Van Halen's "Eruption." I can play the solo for "Free Bird," but I'm no Eddie Van Halen.

All IEEE-HKN members may submit a completed profile of him/herself or someone else. Please submit the profile questionnaire to info@hkn.org. You can also submit suggestions for profiles by nominating names to info@hkn.org.