Creating a common language provides one of a social enterprise’s defining moments, be it a technical, scientific, or humanistic venture. The language’s most obvious utility derives from its allowing a field’s participants to communicate among themselves. Its importance extends beyond this function, however.

More than a neutral medium through which we exchange information, language serves to identify the people who participate in an enterprise and, by its arcanism when related to the common language, sets them apart and contributes to the group identity’s creation. Language establishes a discipline’s accepted ontology, constrains and directs its admissible discourses, and establishes and enforces its orthodoxy. Further, through its etymology, language codifies the relation between a discipline, its history, and the history of other disciplines.

Languages establish their own ontologies and taxonomies, making exact translations of a word nearly impossible. A language also establishes the relations of power and solidarity inside a discipline by identifying roles and determining the statements pertinent to a given role.

LANGUAGE AND COMPUTING

When used by computing professionals—who these days have great social visibility—language also contributes to outsiders’ perception of the direction taken by the areas of society that computers influence. It also determines whether the computing dialect will enter the everyday language and, if so, how future generations will perceive our profession.

How is our creation of language progressing? Not well. Our bizarre etymological choices show that either we don’t know how to create good words or we do know, but don’t think this activity important enough to grant it much attention.

For the past three centuries, scientists have understood that their discipline’s practitioners should have a broad cultural background, which would partly manifest in the words they coin. Words like electron, from the Greek elektra, show a certain proclivity for the classics and a gusto for the creation of beautiful terms.

The computing profession, at least in the US, seems an exception to the rule that people with a university degree should first and foremost be well educated. Worse, the words and expressions our profession creates clearly indicate the scarce sense of history that permeates the engineering world.

INITIALISMS

The most common indicator of our poor lexical literacy comes from the proliferation of acronyms. I see each acronym as a mark of defeat, a surrender of lexicon to the easy play of initials. The few brilliant acronyms aside, most of these terms provide meaningless initialisms. Assemblies of letters such as HTML, VRML, SQL, and PDA look more like noise on a transmission line than something belonging in any human language. Worse, some of these assemblies do not even form correct acronyms: XML should be EML given that it stands for Extensible Markup Language.

These initialisms also reveal a surprising Anglocentrism in a discipline that preaches global communication. English speakers routinely pronounce soundless assemblies of letters, like HTML, by spelling them out. Speakers of Latin languages always create acronyms that can be pronounced as words. Such speakers would view an entity such as HTML as an alien and unpronounceable blurb.

PHONETIC ETYMOLOGY

The computing industry’s curious etymological notions often appear to follow a phonetic rather than logical criterion. Thus we see disgraceful terms like Wi-Fi, which stands for wireless fidelity but evidently derives from Hi-Fi—except that “Hi-Fi” stands for “high fidelity.” Elementary logic dictates that Wi-Fi should thus stand for “wide fidelity,” an expression that makes no sense.

In addition to pseudo-acronyms like XML, we get silly terms like applet—even though application comes from the

Continued on page 110
Latin *applicare.* Following this logic, had the electron been discovered today, it would likely be known as a *chargetlet.*

Other words, while not bad per se, have been used to exhaustion and, worse, with a connotation that radicalizes the acronym’s Anglocentrism into an adamant Americanocentrism. Java provides a typical example. Acceptable as a programming language, it becomes uncomfortably cloying when its advocates build a whole pretend coffee culture around it. After being bombarded with JavaBeans, Moka, Java Espresso, HotJava, Visual Café, Java Café, Café con leche, and so on, I simply want to scream.

Even worse, the connotation of Java that relates it to coffee exists only in American popular culture. The rest of the world thinks of Java as an island in Indonesia, southeast of Sumatra. Alternatively, some archeologists who encounter the term might think of the Pleistocene hominid known as Java man. To see Java generate a plethora of words related to coffee must cause many people outside the US some serious head scratching.

Other words make more sense, but are being inexplicably abandoned in favor of less fitting terms. Many system administrators don’t even call a computer a computer anymore, preferring the incorrect and generic term *box,* as in “We installed a Linux box today.” I can’t help wonder what they call the cardboard container in which the manufacturer shipped that Linux box.

Some words reveal a certain and possibly subliminal recognition of history. However, as often occurs in advanced technology, this recognition is superficial, inadequate, and sees history more as a catalog of funny, outdated curiosities than as a cultural and social process in which we and our technology participate. Often, a curious love for military terminology accompanies this *historicity.* I must confess that every time I attend an all-hands meeting, I instinctively think of reaching for my life jacket and awaiting the whistle announcing that the admiral has boarded.

When not busy fighting imaginary wars, our terms take a wicked pleasure in reminding us of the commodification of the concepts we created and the prominence of computing’s commercial aspect over the cultural, a prominence that so many computing professionals accept uncritically. Terms like *data warehouse* or, worse, *data mart* belong to this group.

To see Java generate a plethora of words related to coffee must cause many people outside the US some serious head scratching.

**OVERRIDE GENERALIZATION**

In other cases, overgeneralization makes the problems worse. The word *e-mail* refers to messages transferred through computer networks. The term itself is objectionable, since the most relevant force behind e-mail is a computer system’s organizational structure, not that it works by moving electrons around. This is one of the fundamental ideas behind abstract notions like the Turing machine and *λ*-calculus. Renouncing it by placing the accent on the raw material of computation—the electron—rather than its process, refutes our discipline’s very foundation.

Already troublesome, the term then unduly generalizes to silly notions like e-commerce, e-books, and e-zines. Not merely an incorrect identification of computation with electrons, this usage exerts an objectionable influence on the lexical culture of other areas. Given that we have no terms such as s-commerce for seaborne commerce, or a-commerce for airborne commerce, why should we prefix commerce or any other cultural enterprise with our silly and omnipresent “e-?”

**CROWNING ACHIEVEMENT**

Two contenders, the word *information* and the prefix *cyber-,* share the crown for most irritantly overused and misused term.

**Information overload**

Computer engineering’s fascination with the term “information” originated with Claude Shannon’s famous 1948 article, “A Mathematics Theory of Computation” (cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf). Shannon used the term quite cautiously—omitting it from the article title—and anecdotes remind us he found it troubling. Further, in his introduction, Shannon states that the “semantic aspects of communications are irrelevant to engineering problems,” a statement true for the engineering problems he considered, but not necessarily for the problems the computing industry takes on these days.

Shannon’s followers, unfortunately, didn’t have the cultural sensibility to follow his cautious lead. Since the early 1980s—thanks to some widely popular and maliciously naive technenthusiasts—the public has been told we have entered the *information age.* This age promises to measure society’s wealth by the information it produces and consumes. Information in this sense consists of a largesse of bits, independent of their meaning or value, a cacophony produced by the proverbial army of monkeys tapping on keyboards from which we still await the equivalent of a Shakespearean play.

For our part, we computing professionals only too happily analyzed and attempted to solve the information overload problem, without questioning whether it arose from mere semantic confusion or, more simply, why we might possibly need more information than we could manage. We also quite happily started using the distasteful, overpublicized, and substantially incorrect term *information technology* in lieu of the humble but correct *data processing.*

**Steered wrong**

When Norbert Wiener coined the word *cybernetics* for his discipline of
control in animals and machines, he knew exactly what he was doing. Derived from the Greek *kibernetike techne*, for the art of the steerer, no term could better convey the idea of control.

Yet, when science fiction author William Gibson’s followers began referring to the Internet as *cyberspace*—the structure without a control center par excellence—it seems unlikely that they knew the root term’s true meaning. We should hope that people praising the power of the world’s largest data repository would at least have had the good sense to check the origin of the terms they planned to use.

Again, bad terms are strangely successful: From that first self-contradictory use of the word, a seemingly unlimited supply of cyber offspring emerged, from the nefarious *cybercafé* to the strangely erotic *cybersex*.

O ur scarce etymological literacy, disregard for so many facets of human culture, and arrogance in imposing the despicable terms we create—when combined with our limited writing ability—spawn a sort of collective autism. We are not receptive to what is beyond us, and those outside our professional circles cannot understand a word we say.

This form of autism can be quite dangerous and is, unfortunately, often promoted in our universities’ curricula. Under corporate pressure for a quickly trained workforce, universities forget to be the universal institutions their name implies. Instead, they focus more on teaching students the latest tricks and less on providing them with an education that will create culturally active and socially aware professionals.

I am not naive enough to believe that we will be able to achieve perennial elegance by adding more literature and philosophy courses to today’s desperately technical engineering curricula. But it’s worth a try. After all, language is the material of thought, and speaking clearly and elegantly means, at least to a degree, thinking clearly and elegantly—a capacity for which the computing industry is in dire need.

**Simone Santini** is a project researcher at the University of California at San Diego. Contact him at ssantini@ncmir.ucsd.edu.