

LESS IS MORE (IS LESS)

Francis Sullivan, Editor in Chief



LITERATURE—IN FACT, ALL THE ARTS—ARE CUSTOMARILY IN THE MIDDLE OF SOME “MOVEMENT” THAT PROMISES TO CHANGE THE WHOLE NATURE OF THE ENDEAVOR. RECENT MOVEMENTS INCLUDE HYPERREALISM, SYMBOLISM, DECONSTRUCTIONISM, EXISTENTIALISM, SURREALISM, AND MANY

other “isms.” One interesting literary movement goes by the name of minimalism. In a minimalist work, sentences are short, as are paragraphs. Even minimalist books are short, with very little narrative connective tissue. Most of the story is told via dialogue, which in the best examples, tells the characters’ hidden story and makes the reader care about what happens to them.

But every movement engenders an opposing one. A quick trip around a bookstore reveals that although many recent minimalist works are available, you’ll also find several very long novels, with large casts of characters and numerous interleaving storylines. In such books, there is a great deal more explicit development of the characters and their surroundings, and the storylines are more complicated. A superb example of a recent minimalist novel is Mary Robison’s novel *Why Did I Ever* (Counterpoint Press, 2002); Jonathan Franzen’s wonderful book *The Corrections* (Farrar & Strauss, 2001)—at 592 pages and more than four intertwining storylines—is clearly not an example of minimalism.

Naturally, the so-called opposition between these two styles is an invention of the critics and not the authors, who probably couldn’t care less and might even dislike being identified with a movement. As an illustration, think of the current practice in writing biographies. Some are multivolume tomes—others are extremely short, easy-to-carry, pocket-size books. Several historical figures have had one of each, and a few authors have written one of each.

As you can probably guess, the same minimal–maximal dynamic operates in software styles and hardware architectures. The C language, which started out on the PDP 7 as the B language, had a clean, stripped-down programming model. It grew up at about the same time as PL/I,

which tried to provide many, many different features. In part, at least, Java was a reaction to C++. In fact, one folk-tale claims that the working name for Java was C++-+++. In recent decades, RISC architectures have been important, but so have complicated caching schemes and multilevel hierarchical memories.

Amusingly, there are even minimal–maximal cycles. An early report by Edsger Dijkstra on how to use a stack mentions a program written for the X1 machine. I’m quite sure that he didn’t mean the most recent high-end offering from Cray.

The point I’m making is that the give and take between maximal and minimal is not a bad thing and should not be thought of as conflict or opposition. Each style is needed, if only for the other one to react against and so make progress. The different opinions about how to proceed do not have the force of different religions, although they are sometimes described that way. I can recall being a young researcher working for a senior professor along with a fellow graduate student and friend. I’d noticed that our supervisor often got things wrong and started barking up the wrong tree. What surprised me is that he would keep on barking, even when it was obvious that the approach wasn’t going to work. It wasn’t until my friend pointed it out that I understood that the barking was important, not the particular tree. CS

New Associate Editor in Chief

John Rundle, *CiSE* editorial board member and director of the Center for Computational Science and Engineering at the University of California, Davis, is the magazine’s newest AEIC, effective this issue.