Book Reviews

MARK LAMOURINE

The Essential Turing: The Ideas That Gave Birth to the Computer Age

B. Jack Copeland, ed.
Oxford University Press, 2004, 614 pages
ISBN 978-0-19-825080-7

It's really good to see Alan Turing finally getting his due in the popular media. He's been a large figure in the mathematical foundations of modern computing from the 1930s (along with John von Neumann and Emil Post, to name just a couple) for quite a long time. Despite this, and despite the fact that Turing's work is often glossed in elementary computing texts (who hasn't at least heard of a Turing machine?), the actual papers on which his reputation is based are not often studied by students of computer science or system administration. It's certainly not necessary anymore than it's necessary to read Copernicus, Galileo, or Newton in the original Latin to be able to do physics or calculus, or to read Euclid in Greek to do geometry. For me, though, something draws me to those original texts.

Turing's work contains much more than his wartime work on Enigma and the justifiably well-known "On Computable Numbers." During his life Turing worked on mathematical topics in artificial intelligence and even artificial life, anticipating the discovery of DNA by positing a computational underpinning to the origin and formation of biological structures. Copeland presents 16 publications on these four topics, ranging from peer-reviewed papers, to a letter from Turing and three others at Bletchley Park that was hand delivered to Winston Churchill to request additional resources for their code-breaking work, to personal mail to his mother during his stay at Princeton before the war. In each case, Copeland provides background and context to help the reader fully appreciate the main texts.

Many of the examples and arguments in Turing's essays may seem obscure or dated to someone who is already familiar with lambda calculus (through the use of Lisp or other modern functional programming languages). A number of them have a decidedly mathematical rather than computational bent, which is understandable when you realize that Turing was writing at a time when no real machines existed or were even under development. It remains remarkable to me that Turing and his colleagues, Alonzo Church, von Neumann, Post and others, conceived these ideas entirely in the abstract. When contrasted with today's methods of prototyping and fast-failure, the rigor involved is impressive (at least to a non-mathematician like myself).

The computational and mathematical writings here are presented in essence and more clearly in modern texts. If you are already familiar with Turing and his work through popular media or formal education in computer science and software development, you are unlikely to learn anything essential to your work. But you will gain insight into the range of topics to which Turing contributed and to the times and environment in which he worked as well as the pleasure of working through his original presentations.

I hope you will.

Postscript: If this kind of reading appeals to you, you might also be interested to find that Stephen Hawking has, over the past decade, released several edited volumes containing the foundational works of classical physics, quantum physics, and mathematics, translated into English and annotated, for those, like me, who feel the call to read them: *On the Shoulders of Giants: The Foundations of Physics and Astronomy; The Dreams That Stuff Is Made of: The Most Astounding Papers of Quantum Physics—and How They Shook the Scientific World;* and *God Created the Integers: The Mathematical Breakthroughs That Changed History.*

Drift into Failure

Sydney Dekker
Ashgate Publishing Ltd., 2011; 220 pages
ISBN 978-1-409402221-1

I think the most striking thing I found in *Drift into Failure* was the final section of the first chapter. That section was titled "Why we must not let Drift into Failure become the next folk model." (The previous section was titled "Great title, lousy metaphor.") The entire chapter was a sort of apology, although I think it was meant to set a framework for the rest of the book.

You see, the book is about how to think about failure, and more precisely, how to think about and analyze events in complex (nonlinear, to use a mathematical term) systems. The first and most important feature of these systems is that they will exhibit unpredictable behavior at times. This is the very nature of complexity, which brings us to the title of the first chapter as a whole (I'm working my way out of the Russian doll I built): "Failure Is Always an Option." If you're looking for a way to eliminate failure, you're reading the wrong book. Or, more significantly, you're doomed to fail, and you should understand how the world really works and pick an achievable goal: understanding how failures happen, looking for the human behaviors that increase

the likelihood of failure (they're not what you would think), and knowing when *not* to waste money "solving" a problem that will never happen again.

Throughout the book, Dekker seems to be aware of the tug of human nature. This is explicit in what he writes but also seems to influence how he writes it. We, modern humans, both by nature (psychology) and culture (the legacy of Descartes and Newton), expect the world to work in a predictable, mechanical, linear way. Asking us to give up the certainty of the Clockwork Universe is a tough sell. We want to be safe, we want to be in control. The argument too is tough: "You want to be certain? You can't be certain, give up." We all answer, "Of course we can! Watch," and we find The Part That Failed and say, "There! That proves I can." Dekker is in the position of trying to prove a negative, to show that while you can always isolate "the cause of failure" after the fact, you cannot in principle prevent all failures by eliminating all points of failure. He knows this and is careful never to offer "the solution."

It's scary to realize that we are not in control *in the way we want to be.* Dekker's argument is that we have two choices: ignore the fact that we're building and depending on complex systems and continue to waste time and effort trying in vain to be 100% safe, or accept that failure is inevitable, but learn to minimize it systematically rather than reductively.

Dekker is trying to show that what we get through our reductionist impulses isn't what we think it is. In that quest he lays out a series of well-known catastrophic failures of technology and analyzes the analysis of the failure and response to the findings. These failures range from a single point mechanical failure that brings down an airliner to the systemic collapse of Enron. Each resulted at its root not from some point failure, but from a series of small, localized, apparently rational decisions that, when seen from a higher scale and in light of the now-apparent flaw, look reckless or even criminal. With each example, he comments on how the seed of a response that would have avoided the failure was already in place, but was minimized or ignored.

Dekker's conclusion is that we, as a society, must change. We must learn to accept risk and failure and respond not by punishing the whistleblowers and the outliers who raise flags before failures, but by encouraging them and listening to them. He advocates creating businesses and other social structures where variety and diversity are accepted, welcomed, and rewarded, because these produce resilient systems. This is a message that has been espoused and championed in the last decade in the software development and service industries as DevOps and Agile methods. More recently, more mainstream businesses have picked up the banner and are finding that, when well done and used appropriately, these methods can work.

There are also cases of both misuse and of failure even when these methods are applied appropriately. The whole point of the book is that failure is inevitable, but that risk is manageable. I think Dekker's reserved tone comes from his understanding of how human nature and modern media, with their two-sides-to-everything mentality, will misrepresent his ideas and lead to a misguided and doomed popular movement akin to the common pop culture abuse of the terms of evolution in places where it just doesn't apply.

For someone able to make a close and careful reading, Dekker will help create a framework with which to begin thinking and working to understand and (as much as is possible) control complex systems in work and in real life. I'm not sure he'll be able to convince the general public, as wedded as it is to a reductionist world model and as insistent on Keeping Me Safe and Finding Someone To Blame as it is. I can only hope.