

Alex Groce (agroce@gmail.com), Northern Arizona University

John Crowley's *Engine Summer* is a novel that feels less like a traditional narrative and more like a cryptic program written in a language only partially understood—a whispered incantation from a future past, delivered in prose both luminous and elusive. At its surface, it tells the story of Rush That Speaks, a young man journeying through a post-technological world where memories and stories are the closest things to data, and where truth is a shape-shifting entity, woven out of voices layered like code commits in a sprawling, fractured archive.

This is a book about storytelling as a kind of software: recursive, iterative, fragile yet enduring. The society Crowley conjures values clarity of narrative, but *Engine Summer* itself resists straightforward decoding. Its structure mirrors the very nature of debugging a sprawling, poorly documented codebase—each chapter a module revealing hints, each character a function with hidden side effects. Meaning emerges slowly, through inference, pattern recognition, and a willingness to dwell in ambiguity.

What makes *Engine Summer* so compelling to those who write code and build systems is how it captures the tension between structure and entropy, between what can be captured in language and what slips through the cracks. The world Crowley builds is one where technology has devolved into myth, and the act of telling true stories is itself revolutionary. This aligns uncannily with the programmer's paradox: to impose order on chaos, yet always confront the limits of that order.

The novel's archival framing—voices stacked within voices, stories nested in stories—feels like a persistent data structure or a version-controlled repository chronicling not just events but evolving perspectives. In this, Crowley anticipates modern concerns about how history, memory, and information intertwine, and how each retelling reprograms the narrative's meaning.

*Engine Summer* does not offer neat answers or triumphant resolutions. Instead, it invites the reader to inhabit the slow unfolding of a world both strange and familiar, a meditation on how we construct meaning, how we preserve knowledge, and ultimately, why we write at all. For anyone who has ever wrestled with complex systems, tangled codebases, or the elusive architecture of stories, Crowley's novel is a haunting and rewarding journey—a reminder that every program we write is, in its way, a story told by candlelight.

...

I didn't write the above column. I asked ChatGPT to, after reading my other published columns, devise a new column. It first put forth a smaller, less interesting, version of my *Metamagical Themas* column. Then it produced a column on Waldrop's *The Dream Machine* that was not half bad; if I'd actually read the book and could confirm that it was a good summary, I'd be willing to use it in a pinch. Then it asked if I wanted a column on a work of fiction, and I responded that I did, and this is what it produced (approximately; details are not an LLM forte, it missed that every column starts with the author's name then the book's name, an invariable

structure that is, at this point, after this many columns, a rule and not a coincidence: the fourteen lines of a sonnet; this is I suspect typical of using LLMs for more ambitious coding tasks: it can mostly do the job, sometimes, but the devil is in the details and LLMs are still really bad at certain details, and in the world of software where so much depends upon details...).

What is this “column”? I am half-inclined to think it’s a parody, and a rather cruel one. The LLM has “intuited” that I love the fiction of John Crowley (and it’s right) and that the basic *modus operandi* of Passages is often to take a work I love for other reasons (best example: John McPhee’s work on deep-time geology) and abuse metaphor to make it “about” software engineering and code. It has my number. It may have your number, too.