

CALIFORNIA



Debugging the Novel

By Alastair Boone

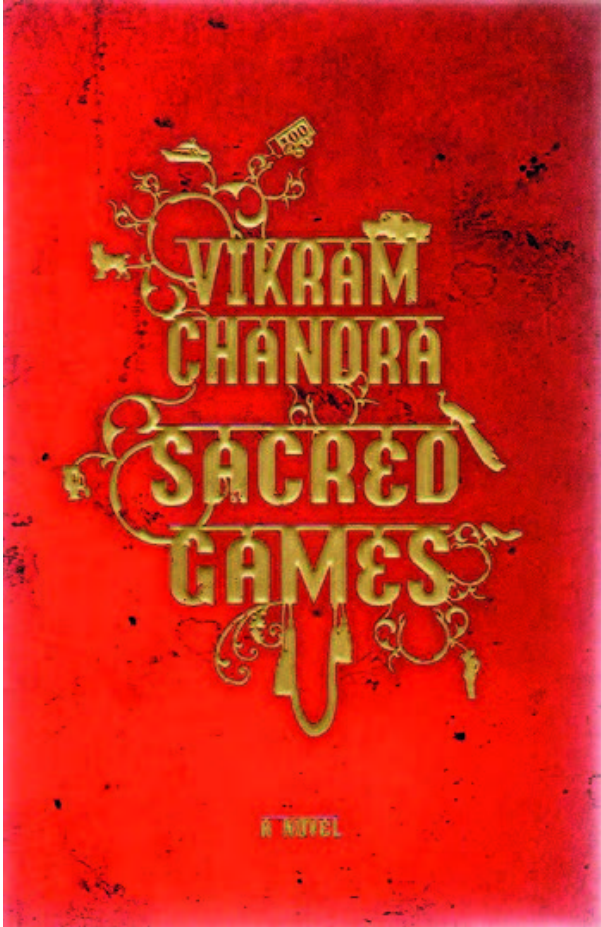
When Vikram Chandra started writing his best-selling novel, [Sacred Games](http://www.vikramchandra.com/publications/sacred-games/) (<http://www.vikramchandra.com/publications/sacred-games/>) (2006), he knew it was going to be a big book. And he was right: All told, the novel is 947 pages, includes over 100 characters, and spans a 60-year timeline. To make the writing process smoother, Chandra set out to find a software program he could use to store, organize, and keep track of the details of his novel. But no off-the-shelf program met his needs.

There are a handful of word processing programs designed specifically for writers—Scrivener, Ulysses, and FocusWriter are three popular ones. Unlike Microsoft Word, which is considered a “linear” program in which pages of writing progress continuously and chronologically, these writing programs allow authors to write in movable chunks. For example, Scrivener uses a corkboard-like layout in which individual notecards on the screen represent chapters or characters. Writers can click on these notecards to expand and edit them, as well as drag-and-drop them into a new order to represent the progression of a novel as it changes over the course of writing. Such software provides a myriad of other features, as well—including the ability to store notes and research—that have proven indispensable to many professional writers.

However, Chandra wanted something more robust to help him track *Sacred Games*’s extensive detail—a program with a nearly human awareness of the story’s minutiae, so the writer doesn’t have to. In short, he was looking for a smart word processor that lends artificial intelligence to the wisdom of a human novelist.

“I looked around, sure that somebody must have built this thing already,” said Chandra, a creative writing professor at Berkeley. But soon he found that, in fact, nobody had built quite what he was looking for.

So Chandra took up the challenge himself. His attempt to improve this system is a word processing program called *Granthika*—a Sanskrit word that translates as “narrator, relater [or] ‘one who understands the joints or divisions of time.’” And the name fits. But when it is released (projected to be late next year), *Granthika* will do more than simply store the working text of a novel in progress. Like a narrator, the software will attend to the meaning of text, as well. For Chandra, this is what the ideal word processor looks like: a framework for storing text, yes; but also an active assistant that can interpret the novel in a way that existing programs cannot.



What does this mean, exactly? Well, let’s say you were to write a novel whose protagonist was born in 1994. If you were to then claim that she turned 21 just 16 years later in 2010, *Granthika* would flag your miscalculation. As you write, the program can construct a timeline around your story, calling out chronological contradictions as they arise.

Future versions of *Granthika* will be able to handle more than linear time. For example, if your novel’s protagonist were a vegetarian and you wrote that she ate a ham sandwich, *Granthika* would flag the inconsistency. “We’re inventing this entirely new way to store not just text but knowledge, and attach them to each other,” Chandra says.

Just as a human reader would read the word “ham” and think “meat,” so too would *Granthika*. Furthermore, *Granthika* will be able to apply first-order logic to the entities in a novel. So if you wrote “John married Jane,” the program would deduce that Jane’s father is now John’s father-in-law.

Authors have obviously found success without the help of software. William Faulkner famously penciled the outline for his 1954 novel, *A Fable*, onto the white walls of the study in his Mississippi home. Nearly 50 years later, J.K. Rowling outlined *Harry Potter and the Order of the Phoenix* using a ballpoint pen to make a spreadsheet on a single sheet of binder paper—listing chapter titles in blue ink from left to right along the top, and a month-by-month timeline down the left side.

Like the pen-and-paper outline, existing software requires the author to make manual updates throughout the writing process. There is still the problem, Chandra says, “that every time you make a change in a place or date, you have to make sure you go through your manuscript and make the appropriate changes. Even though you’re handling your data digitally, you’re not that far from Faulkner drawing the plot for his novel on his wall in Mississippi.”

Not that Faulkner’s primitive outline failed him. On the contrary, *A Fable* won both the Pulitzer Prize for Fiction and the National Book Award for Fiction. However, authors make mistakes all the time, sometimes severe enough to if not unravel a piece of fiction, then at least compromise its integrity. And the mistakes live

in perpetuity. For example, in Arthur Conan Doyle's Sherlock Holmes series, sidekick Watson's first name changes from John to James. And in *Julius Caesar*, Shakespeare writes that the "clock hath stricken three," even though the play takes place in ancient Rome, which didn't yet have mechanical clocks—let alone clocks that tolled.

Chandra insists that the modern novelist should be able to avoid such mistakes. "There's no reason if we're reading digital books that we have to reproduce the scroll," he says, referring to the ebook—"which is essentially what we've got."

Academics have been struggling for decades to improve digital text. In 1987, a group of scholars from around the world founded the Text Encoding Initiative (TEI), an international organization dedicated to developing a standard for the code that governs a digital text. These scholars have been working for 30 years to figure out how to represent text so that it is readable to both a human and a computer. And they have had some success. Scholars have figured out how to mark up digital text so that certain things are searchable, beyond simply determining how many times a given word appears. Inferring meaning, context, or narrative nuance is a lot harder.

In the digital version of *Hamlet*, for example, you can type "Ophelia" into the search bar, and the e-reader will flag every time the word appears. But asking the program to identify which character speaks the most, or whether someone is talking to or about Ophelia, is not likely to be met with success. Now, consider getting a computer to understand an evolving text—one to which additions and revisions are being made—and you can start to understand the sort of challenge that Granthika has taken on.

Another challenge facing Granthika (or any word processor, for that matter) is that writers are a particularly idiosyncratic bunch. In a 2013 [New Yorker](https://www.newyorker.com/magazine/2013/01/14/structure) piece titled "Structure," (<https://www.newyorker.com/magazine/2013/01/14/structure>) John McPhee described his own unique process of outlining a story—one that grew from decades of arranging 3-by-5-inch notecards upon a standard sheet of plywood sitting atop two sawhorses in his office.

In 1984, McPhee met a tech worker at Princeton who wrote a computer program to mimic McPhee's notecard-shuffling process. Although he uses a version of this program to this day, McPhee is the first to admit the singularity of his method. "One size fits one," he wrote.

And while some writers find word processing software to be an indispensable tool, others view the effort it takes to master them a waste of valuable time.

"As a writer myself, I know how annoying it is when a tool you're using calls attention away from the work you're trying to do," Chandra explains. His team is working to combat this aggravation by hiring a user-experience designer, and consulting a group of professional writers to provide input on how Granthika should work. Whether Granthika will aid or annoy remains to be seen. But Ashwin Sanghi—one of Granthika's advisors and among India's best-selling English fiction authors—is hopeful.

“I would love to have a system that keeps track of characters, locations, events, and timelines *as I write*,” Sanghi says. “I think that Granthika simply needs to develop the product and put it out there for writers to wonder how they were managing without it.”

Although Chandra is not the first person to address these problems, he may be the best equipped to solve them. Not just a novelist, he has a long history as a programmer, as well.

Growing up in India, Chandra became enchanted at a young age with reading and writing. According to his best-selling work of nonfiction, *Geek Sublime* (2014), he published his first piece of fiction when he was 12.

“I was a stereotypically budding writer,” he reveals, “thickly bespectacled, shy, bad at cricket, worse at field hockey.” But when his fellow students started to read and appreciate his writing, he found an identity that stuck: “I had found a way to be in the world, to be of it.”

Thus began Chandra’s writing career. He moved to America as a college sophomore to attend Pomona College in Claremont, where he graduated *magna cum laude* in English in 1984. Afterward, he briefly enrolled in Columbia University’s film school before deciding to pursue an MFA in creative writing instead. The following year, he left Columbia to attend the writing seminar at Johns Hopkins University before transferring to the University of Houston to complete his degree. But fiction wasn’t the only thing he was writing; during these years, he also started writing code.



Vikram Chandra / Aubrielle Hvolboll

In an effort to support himself through graduate programs, Chandra took a job as a typist. As it turned out, he could do a lot more than just type. One thing led to another, and he soon became obsessed with computers—learning everything from manipulating software to defragging hard drives. He developed a reputation with friends and strangers all over Houston.

“By now I was walking around campus with 800-page computer manuals tucked under my arm,” he writes in *Geek Sublime*. The learning curve was as difficult as it was exciting. “This particular landscape of American programming is one to which I am a foreigner twice over,” Chandra writes. “I am a writer from India, but I’ve worked professionally as a programmer in the United States. Fiction has been my vocation, and code my obsession.”

With his computer science acumen, he was able to put himself through graduate school without taking out any loans.

In the creative world of writing, it is notable to encounter an author who is also a coder. In *Geek Sublime*, Chandra explains, “Whenever I tell one of my fellow authors that I supported myself through the writing of my first novel by working as a programmer and a computer consultant, I evoke a response that mixes bemusement, bafflement, and a touch of awe, as if I’d just said that I could levitate.”

It certainly captured the attention of fellow author Paul Vidich one evening in 2013 at the Middlebury College Bread Loaf Writers’ Conference in Vermont. After hearing Chandra speak about *Geek Sublime*, Vidich said, “I was particularly struck that this one man’s mind could easily and expertly inhabit two distinct mindsets: the orderly world of code, and the messy world of fiction.”

At the time the two men met, Granthika had become an obsession for Chandra, although he was still stuck on many of the same questions as the scholars working on the TEI—such as how to attach semantic meaning to text despite the problems of inline XML-based markup and its inherently hierarchical syntax. Chandra was particularly excited because he had recently experienced what felt like an epiphany about how he might work around these problems, and all he could do was talk about it. After listening to him, Vidich encouraged Chandra to write a proposal. “As a novelist, I immediately recognized how a software program could help tame the great unruly bear,” Vidich said, alluding to Tolstoy’s description of the novel.

Vidich’s encouragement was a call to action for Chandra, who wrote the proposal soon after their discussion. In the months that followed, he searched the Internet for inspiration—ideas or programming software that might help bring Granthika’s concepts to life. Then, one October evening in 2015, he stumbled upon HyperGraphDB—an “open-source data storage mechanism” founded by Borislav Iordanov. Chandra was intrigued by the model’s knowledge-management capabilities, so he emailed Iordanov with a couple of questions. Iordanov replied that same evening, signing off with, “I’d be curious to learn more about your project.... Would you care to share specifics?”

When he first read Chandra’s proposal, Iordanov “became very excited and dreamy. After many years in the field, it is very rare that I come across an idea that I find genuinely original,” he recalls. Chandra and Iordanov have seemingly very different professional backgrounds, but they became fast friends. The two officially formed the company just three months after they “met” online. (They actually met in person six months later.)

To bring his idea to full fruition, Chandra knew he would need someone with more coding experience than he himself had. A career programmer, Iordanov was perfect for the job. “The first time we talked, he instantly knew what I was talking about, and he could practically see a way of taking the idea and executing it in full,” Chandra said.

Iordanov has worked in software development for more than 20 years. Though his work is largely technical, he has always been interested in the humanities, too. “Even though my background is in computer science proper, I’ve always had a keen interest in philosophical puzzles related to language and meaning,” Iordanov says. “Language has meaning insofar as it generates a computational process in a host—machine or biological—and the meaning is that very process.”

This, the way that language—all language—inspires meaning, is at the heart of Granthika. Written language, after all, is more than strings of characters arranged into a grammatically determined order. Rather, each word has a distinct purpose in furthering the progress of the novel.

Code, too, is its own language, bound by syntax and semantics and rules that have evolved over time. And just as authors make mistakes, so too do programmers. Take JavaScript, for example, which Chandra says is riddled with problems.

“You can dream of order,” Chandra says, “but you’re doing it in a very constant condition of human error and bugginess.”

JavaScript (not to be confused with a different language called Java) is the world’s most widely used programming language—it is sometimes referred to as the *lingua franca* of the Web. But that doesn’t mean it’s state-of-the-art. In fact, it was developed in haste in 1995 by a Netscape employee named Brendan Eich, who was given only ten days to write it. Among developers, JavaScript has a reputation for being sloppy, irrational, and difficult to use.

“It’s a nightmare,” Chandra says. “We all like to think—and especially programmers—that we live in this world of rationality and order, but we’re doing this stuff within this really horrific tool.”

Chandra and Iordanov, like other programmers, have no choice but to use JavaScript, now a fundamental building block for even the simplest online tasks, to build Granthika’s user interface. Although Granthika will start out as a desktop application, using JavaScript will allow them to easily develop Web-based applications in the future.

“You can dream of order,” Chandra says, “but you’re doing it in a very constant condition of human error and bugginess.”

The goal is to build Granthika in such a way that it will be easy for any writer to understand—the code should be elegant to a programmer, and the interface simple for the novelist. If it’s successful, the software could change the way the novel is composed. In Chandra and Iordanov’s wildest dreams, Granthika could even change the future of literature.

“There should be several consequences of a program like Granthika,” says Ashwin Sanghi. “For one, it should reduce the time spent by writers on checking, rechecking, and rewriting their work. It should also reduce the work of fact checkers and copyeditors. It should allow writers to get on with ... telling stories far more complex than the ones they have told so far.”

And if Chandra has his way, the program will eventually change the whole experience of reading fiction. Remember hypertext fiction that let readers pick their own narrative path through a story? It didn’t catch on, partly because technical limitations complicated the tasks of writing and keeping track of all those separate narrative threads. In theory, Granthika will do all that.

“That’s the really ambitious science-fiction-y part of the project ... what we’re really talking about, is a dream that people have had for the last couple of decades,” Chandra explains, “this idea of the book as program.”

Alastair Boone '16 is a writer from San Francisco and former California intern. She currently lives in Washington, D.C., where she is an editorial fellow for CityLab.

From the [Fall 2017 Bugged](#) ([/california-magazine/fall-2017-bugged](#)) issue of California.

Filed under: [Arts + Letters](#) ([/california-magazine/topic/arts-letters](#)) [Innovation](#) ([/california-magazine/topic/innovation](#))

Related topics: [University of California](#) ([/california-magazine/topic/university-california](#)) [California magazine](#) ([/california-magazine/topic/california-magazine](#)) [UC Berkeley](#) ([/california-magazine/topic/uc-berkeley](#)) [alumni](#) ([/california-magazine/topic/alumni](#)) [Cal](#) ([/california-magazine/topic/cal](#)) [Granthika](#) ([/california-magazine/topic/granthika](#)) [Vikram Chandra](#) ([/california-magazine/topic/vikram-chandra](#)) [Sacred Games](#) ([/california-magazine/topic/sacred-games](#)) [narrator](#) ([/california-magazine/topic/narrator](#)) [Harry Potter and the Order of the Phoenix](#) ([/california-magazine/topic/harry-potter-and-order-phoenix](#)) [J.K. Rowling](#) ([/california-magazine/topic/jk-rowling](#)) [William Faulkner](#) ([/california-magazine/topic/william-faulkner](#)) [Mississippi](#) ([/california-magazine/topic/mississippi](#)) [A Fable](#) ([/california-magazine/topic/fable](#)) [Pulitzer Prize](#) ([/california-magazine/topic/pulitzer-prize](#)) [Julius Ceasar](#) ([/california-magazine/topic/julius-ceasar](#)) [Text Encoding Initiative](#) ([/california-magazine/topic/text-encoding-initiative](#)) [Ophelia](#) ([/california-magazine/topic/ophelia](#)) [Hamlet](#) ([/california-magazine/topic/hamlet](#)) [John McPhee](#) ([/california-magazine/topic/john-mcphee](#)) [novel structure](#) ([/california-magazine/topic/novel-structure](#)) [New Yorker](#) ([/california-magazine/topic/new-yorker](#)) [Pomona College](#) ([/california-magazine/topic/pomona-college](#)) [Ashwin Sanghi](#) ([/california-magazine/topic/ashwin-sanghi](#)) [Claremont](#) ([/california-magazine/topic/claremont](#)) [Johns Hopkins University](#) ([/california-magazine/topic/johns-hopkins-university](#)) [University of Houston](#) ([/california-magazine/topic/university-houston](#)) [Geek Sublime](#) ([/california-magazine/topic/geek-sublime](#)) [India](#) ([/california-magazine/topic/india](#)) [computer science](#) ([/california-magazine/topic/computer-science](#)) [Middlebury College Bread Loaf Writer’s Conference](#) ([/california-magazine/topic/middlebury-college-bread-loaf-writers-conference](#)) [XML](#) ([/california-magazine/topic/xml](#)) [hierarchical syntax](#) ([/california-magazine/topic/hierarchical-syntax](#)) [Borislav Iordano](#) ([/california-magazine/topic/borislav-iordano](#)) [HyperGraphDB](#) ([/california-magazine/topic/hypergraphdb](#)) [JavaScript](#) ([/california-magazine/topic/javascript](#)) [Netscape](#) ([/california-magazine/topic/netscape](#)) [Brendan Eich](#) ([/california-magazine/topic/brendan-eich](#)) [lingua franca](#) ([/california-magazine/topic/lingua-franca](#)) [Alastair Boone](#) ([/california-magazine/topic/alastair-boone](#)) [fiction](#) ([/california-magazine/topic/fiction](#)) [logic](#) ([/california-magazine/topic/logic](#)) [semantics](#) ([/california-magazine/topic/semantics](#))

Image source: Aubrielle Hvolboll

Share this article: [↗](#)

Add new comment

Your name *

E-mail *

The content of this field is kept private and will not be shown publicly.

Comment *

Notify me via email when new comments are posted on this page.

By submitting this form, you accept the [Mollom privacy policy \(https://mollom.com/web-service-privacy-policy\)](https://mollom.com/web-service-privacy-policy).