EMERGENCE

THE CONNECTED LIVES OF ANTS, BRAINS, CITIES AND SOFTWARE

STEVEN JOHNSON

PENGUIN BOOKS

1

The Myth of the Ant Queen

It's early fall in Palo Alto, and Deborah Gordon and I are sitting in her office in Stanford's Gilbert Biological Sciences building, where she spends three-quarters of the year studying behavioral ecology. The other quarter is spent doing fieldwork with the native harvester ants of the American Southwest, and when we meet, her face still retains the hint of a tan from her last excursion to the Arizona desert.

I've come here to learn more about the collective intelligence of ant colonies. Gordon, dressed neatly in a white shirt, cheerfully entertains a few borderline-philosophical questions on group behavior and complex systems, but I can tell she's hankering to start with a hands-on display. After a few minutes of casual rumination, she bolts up out of her chair. "Why don't we start with me showing you the ants that we have here," she says. "And then we can talk about what it all means."

She ushers me into a sepulchral room across the hallway, where three long tables are lined up side by side. The initial impression is that of an underpopulated and sterilized pool hall, until I get close enough to one of the tables to make out the miniature civilization that lives within each of them. Closer to a Habitrail than your traditional idea of an ant farm, Gordon's contraptions house an intricate network of plastic tubes connecting a dozen or so plastic boxes, each lined with moist plaster and coated with a thin layer of dirt.

"We cover the nests with red plastic because some species of ants don't see red light," Gordon explains. "That seems to be true of this species too." For a second, I'm not sure what she means by "this species"—and then my eyes adjust to the scene, and I realize with a start that the dirt coating the plastic boxes is, in fact, thousands of harvester ants, crammed so tightly into their quarters that I had originally mistaken them for an undifferentiated mass. A second later, I can see that the whole simulated colony is wonderfully alive, the clusters of ants pulsing steadily with movement. The tubing and cramped conditions and surging crowds bring one thought immediately to mind: the New York subway system, rush hour.

At the heart of Gordon's work is a mystery about how ant colonies develop, a mystery that has implications extending far beyond the parched earth of the Arizona desert to our cities, our brains, our immune systems—and increasingly, our technology. Gordon's work focuses on the connection between the microbehavior of individual ants and the overall behavior of the colonies themselves, and part of that research involves tracking the life cycles of individual colonies, following them year after year as they scour the desert floor for food, competing with other colonies for territory, nd—once a year—mating with them. She is a student, in other ords, of a particular kind of emergent, self-organizing system.

Dig up a colony of native harvester ants and you'll almost invariably find that the queen is missing. To track down the colony's matriarch, you need to examine the bottom of the hole you've just dug to excavate the colony: you'll find a narrow, almost invisible passageway that leads another two feet underground, to a tiny vestibule burrowed out of the earth. There you will find the queen. She will have been secreted there by a handful of ladies-in-waiting at the first sign of disturbance. That passageway, in other words, is an emergency escape hatch, not unlike a fallout shelter buried deep below the West Wing.

But despite the Secret Service–like behavior, and the regal nomenclature, there's nothing hierarchical about the way an ant colony does its thinking. "Although *queen* is a term that reminds us of human political systems," Gordon explains, "the queen is not an authority figure. She lays eggs and is fed and cared for by the workers. She does not decide which worker does what. In a harvester ant colony, many feet of intricate tunnels and chambers and thousands of ants separate the queen, surrounded by interior workers, from the ants working outside the nest and using only the chambers near the surface. It would be physically impossible for the queen to direct every worker's decision about which task to perform and when." The harvester ants that carry the queen off to her escape hatch do so not because they've been ordered to by their leader; they do it because the queen ant is responsible for giving birth to all the members of the colony, and so it's in the colony's best interest—and the colony's gene pool—to keep the queen safe. Their genes instruct them to protect their mother, the same way their genes instruct them to forage for food. In other words, the matriarch doesn't train her servants to protect her, evolution does.

Popular culture trades in Stalinist ant stereotypes—witness the authoritarian colony regime in the animated film *Antz*—but in fact, colonies are the exact opposite of command economies. While they are capable of remarkably coordinated feats of task allocation, there are no Five-Year Plans in the ant kingdom. The colonies that

Gordon studies display some of nature's most mesmerizing decentralized behavior: intelligence and personality and learning that emerges from the bottom up.

I'm still gazing into the latticework of plastic tubing when Gordon directs my attention to the two expansive white boards attached to the main colony space, one stacked on top of the other and connected by a ramp. (Imagine a two-story parking garage built next to a subway stop.) A handful of ants meander across each plank, some porting crumblike objects on their back, others apparently just out for a stroll. If this is the Central Park of Gordon's ant metropolis, I think, it must be a workday.

Gordon gestures to the near corner of the top board, four inches from the ramp to the lower level, where a pile of strangely textured dust—littered with tiny shells and husks—presses neatly against the wall. "That's the midden," she says. "It's the town garbage dump." She points to three ants marching up the ramp, each barely visible beneath a comically oversize shell. "These ants are on midden duty: they take the trash that's left over from the food they've collected—in this case, the seeds from stalk grass—and deposit it in the midden pile."

Gordon takes two quick steps down to the other side of the table, at the far end away from the ramp. She points to what looks like another pile of dust. "And this is the cemetery." I look again, startled. She's right: hundreds of ant carcasses are piled atop one another, all carefully wedged against the table's corner. It looks brutal, and yet also strangely methodical.

I know enough about colony behavior to nod in amazement. "So they've somehow collectively decided to utilize these two areas as trash heap and cemetery," I say. No individual ant defined those areas, no central planner zoned one area for trash, the other for the dead. "It just sort of happened, right?"

Gordon smiles, and it's clear that I've missed something. "It's better than that," she says. "Look at what actually happened here: they've built the cemetery at exactly the point that's furthest away from the colony. And the midden is even more interesting: they've put it at precisely the point that maximizes its distance from both the colony *and* the cemetery. It's like there's a rule they're following: put the dead ants as far away as possible, and put the midden as far away as possible without putting it near the dead ants."

I have to take a few seconds to do the geometry myself, and sure enough, the ants have got it right. I find myself laughing out loud at the thought: it's as though they've solved one of those spatial math tests that appear on standardized tests, conjuring up a solution that's perfectly tailored to their environment, a solution that might easily stump an eight-year-old human. The question is, who's doing the conjuring?

It's a question with a long and august history, one that is scarcely limited to the collective behavior of ant colonies. We know the answer now because we have developed powerful tools for thinking about—and modeling—the emergent intelligence of self-organizing systems, but that answer was not always so clear. We know now that systems like ant colonies don't have real leaders, that the very idea of an ant "queen" is misleading. But the desire to find pacemakers in such systems has always been powerful—in both the group behavior of the social insects, and in the collective human behavior that creates a living city.

Records exist of a Roman fort dating back to A.D. 76 situated at the confluence of the Medlock and Irwell Rivers, on the northwestern edge of modern England, about 150 miles from London. Settlements persisted there for three centuries, before dying out with the rest of the empire around A.D. 400. Historians believe that the site was unoccupied for half a millennium, until a town called Manchester began to take shape there, the name derived from the Roman settlement Mamucium—Latin for "place of the breastlike hill."

Manchester subsisted through most of the millennium as a nondescript northern-England borough: granted a charter in 1301, the town established a college in the early 1400s, but remained secondary to the neighboring town of Salford for hundreds of years. In the 1600s, the Manchester region became a node for the wool trade, its merchants shipping goods to the Continent via the great ports of London. It was impossible to see it at the time, but Manchester—and indeed the entire Lancashire region—had planted itself at the very center of a technological and commercial revolution that would irrevocably alter the future of the planet. Manchester lay at the confluence of several world-historical rivers: the nascent industrial technologies of steam-powered looms; the banking system of commercial London; the global markets and labor pools of the British Empire. The story of that convergence has been told many times, and the debate over its consequences continues to this day. But beyond the epic effects that it had on the global economy, the industrial takeoff that occurred in Manchester between 1700 and 1850 also created a new kind of city, one that literally exploded into existence.

The statistics on population growth alone capture the force of that explosion: a 1773 estimate had 24,000 people living in Manchester; the first official census in 1801 found 70,000. By the midpoint of the century, there were more than 250,000 people in the city proper—a tenfold increase in only seventy-five years. That growth rate was as unprecedented and as violent as the steam engines themselves. In a real sense, the city grew too fast for the authorities to keep up with it. For five hundred years, Manchester had technically been considered a "manor," which meant, in the eyes of the law, it was run like a feudal estate, with no local government to speak of—no city planners, police, or public health authorities. Manchester didn't even send representatives to Parliament until 1832, and it wasn't incorporated for another six years. By the early 1840s, the newly formed borough council finally began to institute public health reforms and urban planning, but the British government didn't officially recognize Manchester as a city until 1853. This constitutes one of the great ironies of the industrial revolution, and it captures just how dramatic the rate of change really was: the city that most defined the future of urban life for the first half of the nineteenth century didn't legally become a city until the great explosion had run its course.

The result of that discontinuity was arguably the least planned and most chaotic city in the six-thousand-year history of urban settlements. Noisy, polluted, massively overcrowded, Manchester attracted a steady stream of intellectuals and public figures in the 1830s, traveling north to the industrial magnet in search of the modern world's future. One by one, they returned with stories of abject squalor and sensory overload, their words straining to convey the immensity and uniqueness of the experience. "What I have seen has disgusted and astonished me beyond all measure," Dickens wrote after a visit in the fall of 1838. "I mean to strike the heaviest blow in my power for these unfortunate creatures." Appointed to command the northern districts in the late 1830s, Major General Charles James Napier wrote: "Manchester is the chimney of the world. Rich rascals, poor rogues, drunken ragamuffins and prostitutes form the moral. . . . What a place! The entrance to hell, realized." De Toqueville visited Lancashire in 1835 and described the landscape in language that would be echoed throughout the next two centuries: "From this foul drain the greatest stream of human industry flows out to fertilize the whole world. From this filthy sewer pure gold flows. Here humanity attains its most complete development and its most brutish; here civilization works its miracles, and civilized man is turned back almost into a savage."

But Manchester's most celebrated and influential documentarian was a young man named Friedrich Engels, who arrived in 1842 to help oversee the family cotton plant there, and to witness firsthand the engines of history bringing the working class closer to self-awareness. While Engels was very much on the payroll of his father's firm, Ermen and Engels, by the time he arrived in Manchester he was also under the sway of the radical politics associated with the Young Hegelian school. He had befriended Karl Marx a few years before and had been encouraged to visit Manchester by the socialist Moses Hess, whom he'd met in early 1842. His three years in England were thus a kind of scouting mission for the revolution, financed by the capitalist class. The book that Engels eventually wrote, *The Condition of the Working Class in England*, remains to this day one of the classic tracts of urban history and stands as the definitive account of nineteenth-century Manchester life in all its tumult and dynamism. Dickens, Carlyle, and Disraeli had all attempted to capture Manchester in its epic wildness, but their efforts were outpaced by a twenty-four-year-old from Prussia.

But *The Condition* is not, as might be expected, purely a document of Manchester's industrial chaos, a story of all that is solid melting into air, to borrow a phrase Engels's comrade would write several years later. In the midst of the city's insanity, Engels's eye is drawn to a strange kind of order, in a wonderful passage where he leads the reader on a walking tour of the industrial capital, a tour that reveals a kind of politics built into the very topography of the city's streets. It captures Engels's acute powers of observation, but I quote from it at length because it captures something else as well—how difficult it is to think in models of self-organization, to imagine a world without pacemakers.

> The town itself is peculiarly built, so that someone can live in it for years and travel into it and out of it daily without ever com-

> ing into contact with a working-class quarter or even with workers—so long, that is to say, as one confines himself to his business affairs or to strolling about for pleasure. This comes about mainly in the circumstances that through an unconscious, tacit agreement as much as through conscious, explicit intention, the working-class districts are most sharply separated from the parts of the city reserved for the middle class. . . .
>
> I know perfectly well that this deceitful manner of building is more or less common to all big cities. I know as well that shopkeepers must in the nature of the business take premises on the main thoroughfares. I know in such streets there are more good houses than bad ones, and that the value of land is higher in their immediate vicinity than in neighborhoods that lie at a distance from them. But at the same time I have never come across so systematic a seclusion of the working class from the main streets as in Manchester. I have never elsewhere seen a concealment of such fine sensibility of everything that might offend the eyes and nerves of the middle classes. And yet it is precisely Manchester that has been built less according to a plan and less within the limitations of official regulations—and indeed more through accident—than any other town. Still . . . I cannot help feeling that the liberal industrialists, the Manchester "bigwigs," are not so altogether innocent of this bashful style of building.

You can almost hear the contradictions thundering against each other in this passage, like the "dark satanic mills" of Manchester itself. The city has built a *cordon sanitaire* to separate the industrialists from the squalor they have unleashed on the world, concealing the demoralization of Manchester's working-class districts—and yet that disappearing act comes into the world without "conscious, explicit intention." The city seems artfully planned to hide its atrocities, and yet it "has been built less according to a plan" than any city in history. As Steven Marcus puts it, in his history of the young Engels's sojourn in Manchester, "The point to be taken is that this astonishing and outrageous arrangement cannot fully be understood as the result of a plot, or even a deliberate design, although those in whose interests it works also control it. It is indeed too huge and too complex a state of organized affairs ever to have been *thought up* in advance, to have preexisted as an idea."

Those broad, glittering avenues, in other words, suggest a Potemkin village without a Potemkin. That mix of order and anarchy is what we now call emergent behavior. Urban critics since Lewis Mumford and Jane Jacobs have known that cities have lives of their own, with neighborhoods clustering into place without any Robert Moses figure dictating the plan from above. But that understanding has entered the intellectual mainstream only in recent years—when Engels paced those Manchester streets in the 1840s, he was left groping blindly, trying to find a culprit for the city's fiendish organization, even as he acknowledged that the city was notoriously unplanned. Like most intellectual histories, the development of that new understanding—the sciences of complexity and self-organization—is a complicated, multithreaded tale, with many agents interacting over its duration. It is probably better to think of it as less a linear narrative and more an interconnected web, growing increasingly dense over the century and a half that separates us from Engels's first visit to Manchester.

Complexity is a word that has frequently appeared in critical accounts of metropolitan space, but there are really two kinds of complexity fundamental to the city, two experiences with very different implications for the individuals trying to make sense of them. There is, first, the more conventional sense of complexity as sensory overload, the city stretching the human nervous system to its very extremes, and in the process teaching it a new series of reflexes—and leading the way for a complementary series of aesthetic values, which develop out like a scab around the original wound. The German cultural critic Walter Benjamin writes in his unfinished masterpiece, *The Arcades Project*:

> Perhaps the daily sight of a moving crowd once presented the eye with a spectacle to which it first had to adapt. . . . [T]hen the assumption is not impossible that, having mastered this task, the eye welcomed opportunities to confirm its possession of its new ability. The method of impressionist painting, whereby the picture is assembled through a riot of flecks of color, would then be a reflection of experience with which the eye of a big-city dweller has become familiar.

There's a long tributary of nineteenth- and twentieth-century urban writing that leads into this passage, from the London chapters of Wordsworth's *Prelude* to the ambulatory musings of Joyce's *Dubliners*: the noise and the senselessness somehow transformed into an aesthetic experience. The crowd is something you throw yourself into, for the pure poetry of it all. But complexity is not solely a matter of sensory overload. There is also the sense of complexity as a self-organizing system—more Santa Fe Institute than Frankfurt School. This sort of complexity lives up one level: it describes the system of the city itself, and not its experiential reception by the city dweller. The city is complex because it overwhelms, yes, but also because it has a coherent personality, a personality that self-organizes out of millions of individual decisions, a global order built out of local interactions. This is the "systematic" complexity that Engels glimpsed on the boulevards of Manchester: not the overload and anarchy he documented elsewhere, but instead a strange kind of order, a pattern in the streets that furthered the political values of

Manchester's elite without being deliberately planned by them. We know now from computer models and sociological studies—as well as from the studies of comparable systems generated by the social insects, such as Gordon's harvester ants—that larger patterns can emerge out of uncoordinated local actions. But for Engels and his contemporaries, those unplanned urban shapes must have seemed like a haunting. The city appeared to have a life of its own.

A hundred and fifty years later, the same techniques translated into the language of software—as in Mitch Resnick's slime mold simulation—trigger a similar reaction: the eerie sense of something lifelike, something organic forming on the screen. Even those with sophisticated knowledge about self-organizing systems still find these shapes unnerving—in their mix of stability and change, in their capacity for open-ended learning. The impulse to build centralized models to explain that behavior remains almost as strong as it did in Engels's day. When we see repeated shapes and structure emerging out of apparent chaos, we can't help looking for pacemakers.

Understood in the most abstract sense, what Engels observed are *patterns* in the urban landscape, visible because they have a repeated structure that distinguishes them from the pure noise you might naturally associate with an unplanned city. They are patterns of human movement and decision-making that have been etched into the texture of city blocks, patterns that are then fed back to the Manchester residents themselves, altering their subsequent decisions. (In that sense, they are the very opposite of the traditional sense of urban complexity—they are signals emerging where you would otherwise expect only noise.) A city is a kind of pattern-amplifying machine: its neighborhoods are a way of measuring and expressing the repeated behavior of larger collectivities—capturing information about group behavior, and sharing that information with the group. Because those patterns are fed back to the community, small shifts in behavior can quickly escalate into larger movements: upscale shops dominate the main boulevards, while the working class remains clustered invisibly in the alleys and side streets; the artists live on the Left Bank, the investment bankers in the Eighth Arrondissement. You don't need regulations and city planners deliberately creating these structures. All you need are thousands of individuals and a few simple rules of interaction. The bright shop windows attract more bright shop windows and drive the impoverished toward the hidden core. There's no need for a Baron Haussmann in this world, just a few repeating patterns of movement, amplified into larger shapes that last for lifetimes: clusters, slums, neighborhoods.

Not all patterns are visible to every city dweller, though. The history of urbanism is also the story of more muted signs, built by the collective behavior of smaller groups and rarely detected by outsiders. Manchester harbors several such secret clusters, persisting over the course of many generations, like a "standing wave in front of a rock in a fast-moving stream." One of them lies just north of Victoria University, at a point where Oxford Road becomes Oxford Street. There are reports dating back to the mid-nineteenth century of men cruising other men on these blocks, looking for casual sex, more lasting relationships, or even just the camaraderie of shared identity at a time when that identity dared not speak its name. Some historians speculate that Wittgenstein visited these streets during his sojourn in Manchester in 1908. Nearly a hundred years later, the area has christened itself the Gay Village and actively promotes its coffee bars and boutiques as a must-see Manchester tourist destination, like Manhattan's Christopher Street and San Francisco's Castro. The pattern is now broadcast to a wider audience, but it has not lost its shape.

But even at a lower amplitude, that signal was still loud enough to attract the attention of another of Manchester's illustrious immigrants: the British polymath Alan Turing. As part of his heroic contribution to the war effort, Turing had been a student of mathematical patterns, designing the equations and the machines that cracked the "unbreakable" German code of the Enigma device. After a frustrating three-year stint at the National Physical Laboratory in London, Turing moved to Manchester in 1948 to help run the university's embryonic computing lab. It was in Manchester that Turing began to think about the problem of biological development in mathematical terms, leading the way to the "Morphogenesis" paper, published in 1952, that Evelyn Fox Keller would rediscover more than a decade later. Turing's war research had focused on detecting patterns lurking within the apparent chaos of code, but in his Manchester years, his mind gravitated toward a mirror image of the original code-breaking problem: how complex patterns could come into being by following simple rules. How does a seed know how to build a flower?

Turing's paper on morphogenesis—literally, "the beginning of shape"—turned out to be one of his seminal works, ranking up their with his more publicized papers and speculations: his work on Gödel's undecidability problem, the Turing Machine, the Turing Test—not to mention his contributions to the physical design of the modern digital computer. But the morphogenesis paper was only the beginning of a shape—a brilliant mind sensing the outlines of a new problem, but not fully grasping all its intricacies. If Turing had been granted another few decades to explore the powers of self-assembly—not to mention access to the number-crunching horsepower of non-vacuum-tube computers—it's not hard to imagine his mind greatly enhancing our subsequent understanding of emergent behavior. But the work on morphogenesis was tragically cut short by his death in 1954.

Alan Turing was most likely a casualty of the brutally homophobic laws of postwar Britain, but his death also intersected with those discreet patterns of life on Manchester's sidewalks. Turing had known about that stretch of Oxford Road since his arrival in Manchester; on occasion, he would drift down to the neighborhood, meeting other gay men—inviting some of them back to his flat for conversation, and presumably some sort of physical contact. In January of 1952, Turing met a young man named Arnold Murray on those streets, and the two embarked on a brief relationship that quickly turned sour. Murray—or a friend of Murray's—broke into Turing's house and stole a few items. Turing reported the theft to the police and, with his typical forthrightness, made no effort to conceal the affair with Murray when the police visited his flat. Homosexuality was a criminal offense according to British law, punishable by up to two years' imprisonment, and so the police promptly charged both Turing and Murray with "gross indecency."

On February 29, 1952, while the Manchester authorities were preparing their case against him, Turing finished the revisions to his morphogenesis paper, and he argued over its merits with Ilya Prigogine, the visiting Belgian chemist whose work on nonequilibrium thermodynamics would later win him a Nobel prize. In one day, Turing had completed the text that would help engender the discipline of biomathematics and inspire Keller and Segel's slime mold discoveries fifteen years later, and he had enjoyed a spirited exchange with the man who would eventually achieve world fame for his research into self-organizing systems. On that winter day in 1952, there was no mind on the face of the earth better prepared to wrestle with the mysteries of emergence than Alan Turing's. But the world outside that mind was conspiring to destroy it. That very morning, a local paper broke the story that the war-hero savant had been caught in an illicit affair with a nineteen-year-old boy.

Within a few months Turing had been convicted of the crime and placed on a humiliating estrogen treatment to "cure" him of his homosexuality. Hounded by the authorities and denied security clearance for the top-secret British computing projects he had been contributing to, Turing died two years later, an apparent suicide.

Turing's career had already collided several times with the developing web of emergence before those fateful years in Manchester. In the early forties, during the height of the war effort, he had spent several months at the legendary Bell Laboratories on Manhattan's West Street, working on a number of encryption schemes, including an effort to transmit heavily encoded waveforms that could be decoded as human speech with the use of a special key. Early in his visit to Bell Labs, Turing hit upon the idea of using another Bell invention, the Vocoder—later used by rock musicians such as Peter Frampton to combine the sounds of a guitar and the human voice—as a way of encrypting speech. (By early 1943, Turing's ideas had enabled the first secure voice transmission to cross the Atlantic, unintelligible to German eavesdroppers.) Bell Labs was the home base for another genius, Claude Shannon, who would go on to found the influential discipline of information theory, and whose work had explored the boundaries between noise and information. Shannon had been particularly intrigued by the potential for machines to detect and amplify patterns of information in noisy communication channels—a line of inquiry that promised obvious value to a telephone company, but could also save thousands of lives in a war effort that relied so heavily on the sending and breaking of codes. Shannon and Turing immediately recognized that they had been working along parallel tracks: they were both code-breakers by profession at that point, and in their attempts to build automated machines that could recognize patterns in audio signals or numerical sequences, they had both glimpsed a future populated by even more intelligence machines. Shannon and Turing passed many an extended lunchtime at the Bell Labs, trading ideas on an "electronic brain" that might be capable of humanlike feats of pattern recognition.

Turing had imagined his thinking machine primarily in terms of its logical possibilities, its ability to execute an infinite variety of computational routines. But Shannon pushed him to think of the machine as something closer to an actual human brain, capable of recognizing more nuanced patterns. One day over lunch at the lab, Turing exclaimed playfully to his colleagues, "Shannon wants to feed not just data to a brain, but *cultural* things! He wants to play music to it!" Musical notes were patterns too, Shannon recognized, and if you could train an electronic brain to understand and respond to logical patterns of zeros and ones, then perhaps sometime in the future we could train our machines to appreciate the equivalent patterns of minor chord progressions and arpeggios. The idea seemed fanciful at the time—it was hard enough getting a machine to perform long division, much less savor Beethoven's Ninth. But the pattern recognition that Turing and Shannon envisioned for digital computers has, in recent years, become a central part of our cultural life, with machines both generating music for our entertainment and recommending new artists for us to enjoy. The connection between musical patterns and our neurological wiring would play a central role in one of the founding texts of modern artificial intelligence, Douglas Hofstadter's *Gödel, Escher, Bach*. Our computers still haven't developed a genuine ear for music, but if they ever do, their skill will date back to those lunchtime conversations between Shannon and Turing at Bell Labs. And that learning too will be a kind of emergence, a higher-level order forming out of relatively simple component parts.

Five years after his interactions with Turing, Shannon published a long essay in the *Bell System Technical Journal* that was quickly repackaged as a book called *The Mathematical Theory of Communication.* Dense with equations and arcane chapter titles such as "Discrete Noiseless Systems," the book managed to become something of a cult classic, and the discipline it spawned—information theory—had a profound impact on scientific and technological research that followed, on both a theoretical and practical level. *The Mathematical Theory of Communication* contained an elegant, layman's introduction to Shannon's theory, penned by the esteemed scientist Warren Weaver, who had early on grasped the significance of Shannon's work. Weaver had played a leading role in the Natural Sciences division of the Rockefeller Foundation since 1932, and when he retired in the late fifties, he composed a long report for the foundation, looking back at the scientific progress that had been achieved over the preceding quarter century. The occasion suggested a reflective look backward, but the document that Weaver produced (based loosely on a paper he had written for *American Scientist)* was far more prescient, more forward-looking. In many respects, it deserves to be thought of as the founding text of complexity theory—the point at which the study of complex systems began to think of itself as a unified field. Drawing upon research in molecular biology, genetics, physics, computer science, and Shannon's information theory, Weaver divided the last few centuries of scientific inquiry into three broad camps. First, the study of simple systems: two or three variable problems, such as the rotation of planets, or the connection between an electric current and its voltage and resistance. Second, problems of "disorganized complexity": problems characterized by millions or billions of variables that can only be approached by the methods of statistical mechanics and probability theory. These tools helped explain not only the behavior of molecules in a gas, or the patterns of heredity in a gene pool, but also helped life insurance companies turn a profit despite their limited knowledge about any individual human's future health. Thanks to Claude Shannon's work, the statistical approach also helped phone companies deliver more reliable and intelligible long-distance service.

But there was a third phase to this progression, and we were only beginning to understand. "This statistical method of dealing with disorganized complexity, so powerful an advance over the earlier two-variable methods, leaves a great field untouched," Weaver wrote. There was a middle region between two-variable equations and problems that involved billions of variables. Conventionally, this region involved a "moderate" number of variables, but the size of the system was in fact a secondary characteristic:

> Much more important than the mere number of variables is the fact that these variables are all interrelated. . . . These problems, as contrasted with the disorganized situations with which statistics can cope, *show the essential feature of organization.* We will therefore refer to this group of problems as those of *organized complexity.*

Think of these three categories of problems in terms of our billiards table analogy from the introduction. A two- or three-variable problem would be an ordinary billiards table, with balls bouncing off one another following simple rules: their velocities, the friction of the table. That would be an example of a "simple system"—and indeed, billiard balls are often used to illustrate basic laws of physics in high school textbooks. A system of disorganized complexity would be that same table enlarged to include a million balls, colliding with one another millions of times a second. Making predictions about the behavior of any individual ball in that mix would be difficult, but you could make some accurate predictions about the overall behavior of the table. Assuming there's enough energy in the system at the outset, the balls will spread to fill the entire table, like gas molecules in a container. It's complex because there are many interacting agents, but it's disorganized because they don't create any higher-level behavior other than broad statistical trends. Organized complexity, on the other hand, is like our motorized billiards table, where the balls follow specific rules and through their various interactions create a distinct macrobehavior, arranging themselves in a specific shape, or forming a specific pattern over time. That sort of behavior, for Weaver, suggested a problem of organized complexity, a problem that suddenly seemed omnipresent in nature once you started to look for it:

> What makes an evening primrose open when it does? Why does salt water fail to satisfy thirst? . . . What is the description of aging in biochemical terms? . . . What is a gene, and how does the original genetic constitution of a living organism express itself in the developed characteristics of the adult?
>
> All these are certainly complex problems. But they are not problems of disorganized complexity, to which statistical methods hold the key. They are all problems which involve dealing simultaneously with a sizable number of factors which are interrelated into an organic whole.

Tackling such problems required a new approach: "The great central concerns of the biologist . . . are now being approached not only from *above,* with the broad view of the natural philosopher who scans the whole living world, but also from *underneath,* by the quantitative analyst who measures the underlying facts." This was a genuine shift in the paradigm of research, to use Thomas Kuhn's language—a revolution not so much in the interpretations that science built in its attempt to explain the world, but rather in the types of questions it asked. The paradigm shift was more than just a new mind-set, Weaver recognized; it was also a by-product of new tools that were appearing on the horizon. To solve the problems of organized complexity, you needed a machine capable of churning through thousands, if not millions, of calculations per second—a rate that would have been unimaginable for individual brains running the numbers with the limited calculating machines of the past few centuries. Because of his connection to the Bell Labs group, Weaver had seen early on the promise of digital computing, and he knew that the mysteries of organized complexity would be much easier to tackle once you could model the behavior in close-to-real time. For millennia, humans had used their skills at observation and classification to document the subtle anatomy of flowers, but for the first time they were perched on the brink of answering a more fundamental question, a question that had more to do with patterns developing over time than with static structure: Why does an evening primrose open when it does? And how does a simple seed know how to make a primrose in the first place?

Alan Turing had played an essential role in creating both the hardware and the software that powered this first digital revolution, and his work on morphogenesis had been one of the first systematic attempts to imagine development as a problem of organized complexity. It is one of the great tragedies of this story that Turing didn't live to see—much less participate in—the extraordinary intellectual flowering that took place when those two paths intersected.

Ironically, Warren Weaver's call to action generated the first major breakthrough in a work that had nothing to do with digital computers—a work that belonged to a field not usually considered part of the hard sciences. In the years after the war, urban planners and government officials had been tackling the problem of inner-city slums with a decidedly top-down approach: razing entire neighborhoods and building bleak high-rise housing projects, ringed by soon-to-be-derelict gardens and playgrounds. The projects effectively tried to deal with the problem of dangerous city streets by eliminating streets altogether, and while the apartments in these new high-rises usually marked an improvement in living space and infrastructure, the overall environment of the projects quickly descended into an anonymous war zone that managed to both increase the crime rate in the area and destroy the neighborhood feel that had preceded them.

In October of 1961, the New York City Planning Commission announced its findings that a large portion of the historic West Village was "characterized by blight, and suitable for clearance, replanning, reconstruction, or rehabilitation." The Village community—a lively mix of artists, writers, Puerto Rican immigrants, and working-class Italian-Americans—responded with outrage, and at the center of the protests was an impassioned urban critic named Jane Jacobs. Jacobs had just spearheaded a successful campaign to block urban-development kingpin Robert Moses's plan to build a superhighway through the heart of SoHo, and she was now turning her attention to the madness of the projects. (The proposed "rehabilitation" included Jacobs's own residence on Hudson Street.) In her valiant and ultimately triumphant bid to block the razing of the West Village, Jacobs argued that the way to improve city streets and restore the dynamic civility of urban life was not to bulldoze the problem zones, but rather to look at city streets that did work and learn from them. Sometime in the writing of what would become *The Death and Life of the Great American Cities*—published shortly after the Village showdown—Jacobs read Warren Weaver's Rockefeller Foundation essay, and she immediately recognized her own agenda in his call for exploring problems of organized complexity.

> Under the seeming disorder of the old city, wherever the old city is working successfully, is a marvelous order for maintaining the safety of the streets and the freedom of the city. It is a complex order. Its essence is intimacy of sidewalk use, bringing with it a constant succession of eyes. This order is all composed of movement and change, and although it is life, not art, we may fancifully call it the art form of the city and liken it to the dance—not to a simple-minded precision dance with everyone kicking up at the same time, twirling in unison and bowing off en masse, but to an intricate ballet in which the individual dancers and ensembles all have distinctive parts which miraculously reinforce each other and compose an orderly whole.

Jacobs gave *Death and Life*'s closing chapter the memorable title "The Kind of Problem a City Is," and she began it by quoting extensively from Weaver's essay. Understanding how a city works, Jacobs argued, demanded that you approach it as a problem from the street level up. "In parts of cities which are working well in some respects and badly in others (as is often the case), we cannot even analyze the virtues and the faults, diagnose the trouble or consider helpful changes, without going at them as problems of organized complexity," she wrote. "We may wish for easier, all-purpose analyses, and for simpler, magical, all-purpose cures, but wishing cannot change these problems into simpler matters than organized complexity, no matter how much we try to evade the realities and to handle them as something different." To understand the city's complex order, you needed to understand that ever-changing ballet; where city streets had lost their equilibrium, you couldn't simply approach the problem by fiat and bulldoze entire neighborhoods out of existence.

Jacobs's book would revolutionize the way we imagined cities. Drawing on Weaver's insights, she conveyed a vision of the city as far more than the sum of its residents—closer to a living organism, capable of adaptive change. "Vital cities have marvelous innate abilities for understanding, communicating, contriving and inventing what is required to combat their difficulties," she wrote. They get their order from below; they are learning machines, pattern recognizers—even when the patterns they respond to are unhealthy ones. A century after Engels glimpsed the systematic disappearing act of Manchester's urban poor, the self-organizing city had finally come into focus.

"Organized complexity" proved to be a constructive way of thinking about urban life, but Jacobs's book was a work of social theory, not science. Was it possible to model and explain the behavior of self-organizing systems using more rigorous methods? Could the developing technology of digital computing be usefully applied to this problem? Partially thanks to Shannon's work in the late forties, the biological sciences had made a number of significant breakthroughs in understanding pattern recognition and feedback by the time Jacobs published her masterpiece. Shortly after his appointment to the Harvard faculty in 1956, the entomologist Edward O. Wilson convincingly proved that ants communicate with one another—and coordinate overall colony behavior—by recognizing patterns in pheromone trails left by fellow ants, not unlike the cyclic AMP signals of the slime mold. At the Free University of Brussels in the fifties, Ilya Prigogine was making steady advances in his understanding of nonequilibrium thermodynamics, environments where the laws of entropy are temporarily overcome, and higher-level order may spontaneously emerge out of underlying chaos. And at MIT's Lincoln Laboratory, a twenty-five-year-old researcher named Oliver Selfridge was experimenting with a model for teaching a computer how to learn.

There is a world of difference between a computer that passively receives the information you supply and a computer that actively learns on its own. The very first generation of computers such as ENIAC had processed information fed to them by their masters, and they had been capable of performing various calculations with that data, based on the instruction sets programmed into them. This was a startling enough development at a time when "computer" meant a person with a slide rule and an eraser. But even in those early days, the digital visionaries had imagined a machine capable of more open-ended learning. Turing and Shannon had argued over the future musical tastes of the "electronic brain" during lunch hour at Bell Labs, while their colleague Norbert Wiener had written a best-selling paean to the self-regulatory powers of feedback in his 1949 manifesto *Cybernetics.*

"Mostly my participation in all of this is a matter of good luck for me," Selfridge says today, sitting in his cramped, windowless MIT office. Born in England, Selfridge enrolled at Harvard at the age of fifteen and started his doctorate three years later at MIT, where Norbert Wiener was his dissertation adviser. As a precocious twenty-one-year-old, Selfridge suggested a few corrections to a paper that his mentor had published on heart flutters, corrections that Wiener graciously acknowledged in the opening pages of *Cybernetics.* "I think I now have the honor of being one of the few living people mentioned in that book," Selfridge says, laughing.

After a sojourn working on military control projects in New Jersey, Selfridge returned to MIT in the midfifties. His return coincided with an explosion of interest in artificial intelligence (AI), a development that introduced him to a then-junior fellow at Harvard named Marvin Minsky. "My concerns in AI," Selfridge says now, "were not so much the actual processing as they were in how systems change, how they evolve—in a word, how they learn." Exploring the possibilities of machine learning brought Selfridge back to memories of his own education in England. "At school in England I had read John Milton's *Paradise Lost,*" he says, "and I'd been struck by the image of Pandemonium—it's Greek for 'all the demons.' Then after my second son, Peter, was born, I went over *Paradise Lost* again, and the shrieking of the demons awoke something in me." The pattern recognizer in Selfridge's brain had hit upon a way of teaching a computer to recognize patterns.

"We are proposing here a model of a process which we claim can adaptively improve itself to handle certain pattern-recognition problems which cannot be adequately specified in advance." These were the first words Selfridge delivered at a symposium in late 1958, held at the very same National Physical Laboratory from which Turing had escaped a decade before. Selfridge's presentation had the memorable title "Pandemonium: A Paradigm for Learning," and while it had little impact outside the nascent computer-science community, the ideas Selfridge outlined that day would eventually become part of our everyday life—each time we enter a name in our PalmPilots or use voice-recognition software to ask for information over the phone. Pandemonium, as Selfridge outlined it in his talk, was not so much a specific piece of software as it was a way of approaching a problem. The problem was an ambitious one, given the limited computational resources of the day: how to teach a computer to recognize patterns that were ill-defined or erratic, like the sound waves that comprise spoken language.

The brilliance of Selfridge's new paradigm lay in the fact that it relied on a distributed, bottom-up intelligence, and not a unified, top-down one. Rather than build a single smart program, Selfridge created a swarm of limited miniprograms, which he called demons. "The idea was, we have a bunch of these demons shrieking up the hierarchy," he explains. "Lower-level demons shrieking to higher-level demons shrieking to higher ones."

To understand what that "shrieking" means, imagine a system with twenty-six individual demons, each trained to recognize a letter of the alphabet. The pool of demons is shown a series of words, and each demon "votes" as to whether each letter displayed represents its chosen letter. If the first letter is *a*, the *a*-recognizing demon reports that it is highly likely that it has recognized a match. Because of the similarities in shape, the *o*-recognizer might report a possible match, while the *b*-recognizer would emphatically declare that the letter wasn't intelligible to it. All the letter-recognizing demons would report to a master demon, who would tally up the votes for each letter and choose the demon that expressed the highest confidence. Then the software would move on to the next letter in the sequence, and the process would begin again. At the end of the transmission, the master demon would have a working interpretation of the text that had been transmitted, based on the assembled votes of the demon democracy.

Of course, the accuracy of that interpretation depended on the accuracy of the letter recognizers. If you were trying to teach a computer how to read, it was cheating to assume from the outset that you could find twenty-six accurate letter recognizers. Selfridge was after a larger goal: How do you teach a machine to recognize letters—or vowel sounds, minor chords, fingerprints—in the first place? The answer involved adding another layer of demons, and a feedback mechanism whereby the various demon guesses could be graded. This lower level was populated by even less sophisticated miniprograms, trained only to recognize raw physical shapes (or sounds, in the case of Morse code or spoken language). Some demons recognized parallel lines, others perpendicular ones. Some demons looked for circles, others for dots. None of these shapes were associated with any particular letter; these bottom-dwelling demons were like two-year-old children—capable of reporting on the shapes they witnessed, but not perceiving them as letters or words.

Using these minimally equipped demons, the system could be trained to recognize letters, without "knowing" anything about the alphabet in advance. The recipe was relatively simple: Present the letter *b* to the bottom-level demons, and see which ones respond, and which ones don't. In the case of the letter *b,* the vertical-line recognizers might respond, along with the circle recognizers. Those lower-level demons would report to a letter-recognizer one step higher in the chain. Based on the information gathered from its lieutenants, that recognizer would make a guess as to the letter's identity. Those guesses are then "graded" by the software. If the guess is wrong, the software learns to dissociate those particular lieutenants from the letter in question; if the guess happens to be right, it *strengthens* the connection between the lieutenants and the letter.

The results are close to random at first, but if you repeat the process a thousand times, or ten thousand, the system learns to associate specific assembles of shape-recognizers with specific letters and soon enough is capable of translating entire sentences with remarkable accuracy. The system doesn't come with any predefined conceptions about the shapes of letters—you train the system to associate letters with specific shapes in the grading phase. (This is why handwriting-recognition software can adapt to so many different types of penmanship, but *can't* adapt to penmanship that changes day to day.) That mix of random beginnings organizing into more complicated results reminded Selfridge of another process, whose own underlying code was just then being deciphered in the form of DNA. "The scheme sketched is really a natural selection on the processing demons," Selfridge explained. "If they serve a useful function they survive and perhaps are even the source for other subdemons who are themselves judged on their merits. It is perfectly reasonable to conceive of this taking place on a broader scale . . . instead of having but one Pandemonium we might have some crowd of them, all fairly similarly constructed, and employ natural selection on the crowd of them."

The system Selfridge described—with its bottom-up learning, and its evaluating feedback loops—belongs in the history books as the first practical description of an emergent software program. The world now swarms with millions of his demons.

Among the students at MIT in the late forties was a transplanted midwesterner named John Holland. Holland was also a pupil of Norbert Wiener's, and he spent a great deal of his undergraduate years stealing time on the early computer prototypes being built in Cambridge at that time. His unusual expertise at computer programming led IBM to hire him in the fifties to help develop their first commercial calculator, the 701. As a student of Wiener's, he was naturally inclined to experiment with ways to make the sluggish 701 machine learn in a more organic, bottom-up fashion—not unlike Selfridge's Pandemonium—and Holland and a group of like-minded colleagues actually programmed a crude simulation of neurons interacting. But IBM was in the business of selling adding machines then, and so Holland's work went largely ignored and underfunded. After a few years Holland returned to academia to get his doctorate at the University of Michigan, where the Logic of Computers Group had just been formed.

In the sixties, after graduating as the first computer science Ph.D. in the country, Holland began a line of inquiry that would dominate his work for the rest of his life. Like Turing, Holland wanted to explore the way simple rules could lead to complex behavior; like Selfridge, he wanted to create software that would be capable of open-ended learning. Holland's great breakthrough was to harness the forces of another bottom-up, open-ended system: natural selection. Building on Selfridge's Pandemonium model, Holland took the logic of Darwinian evolution and built it into code. He called his new creation the genetic algorithm.

A traditional software program is a series of instructions that tells the computer what to do: paint the screen with red pixels, multiply a set of numbers, delete a file. Usually those instructions are encoded as a series of branching paths: do this first, and if you get result A, do one thing; if you get result B, do another thing. The art of programming lay in figuring out how to construct the most efficient sequence of instructions, the sequence that would get the most done with the shortest amount of code—and with the least likelihood of a crash. Normally that was done using the raw intellectual firepower of the programmer's mind. You thought about the problem, sketched out the best solution, fed it into the computer, evaluated its success, and then tinkered with it to make it better. But Holland imagined another approach: set up a gene pool of possible software and let successful programs *evolve* out of the soup.

Holland's system revolved around a series of neat parallels between computer programs and earth's life-forms. Each depends on a master code for its existence: the zeros and ones of computer programming, and the coiled strands of DNA lurking in all of our cells (usually called the genotype). Those two kinds of codes dictate some kind of higher-level form or behavior (the phenotype): growing red hair or multiplying two numbers together. With DNA-based organisms, natural selection works by creating a massive pool of genetic variation, then evaluating the success rate of the assorted behaviors unleashed by all those genes. Successful variations get passed down to the next generation, while unsuccessful ones disappear. Sexual reproduction ensures that the innovative combinations of genes find each other. Occasionally, random mutations appear in the gene pool, introducing complete new avenues for the system to explore. Run through enough cycles, and you have a recipe for engineering masterworks like the human eye—without a bona fide engineer in sight.

The genetic algorithm was an attempt to capture that process in silicon. Software already has a genotype and a phenotype, Holland recognized; there's the code itself, and then there's what the code actually *does.* What if you created a gene pool of different code combinations, then evaluated the success rate of the phenotypes, eliminating the least successful strands? Natural selection relies on a brilliantly simple, but somewhat tautological, criterion for evaluating success: your genes get to pass on to the next generation if you survive long enough to produce a next generation. Holland decided to make that evaluation step more precise: his programs would be admitted to the next generation if they did a better job of accomplishing a specific task—doing simple math, say, or recognizing patterns in visual images. The programmer could decide what the task was; he or she just couldn't directly instruct the software how to accomplish it. He or she would set up the parameters that defined genetic fitness, then let the software evolve on its own.

Holland developed his ideas in the sixties and seventies using mostly paper and pencil—even the more advanced technology of that era was far too slow to churn through the thousandfold generations of evolutionary time. But the massively parallel, high-speed computers introduced in the eighties—such as Danny Hillis's Connection Machine—were ideally suited for exploring the powers of the genetic algorithm. And one of the most impressive GA systems devised for the Connection Machine focused exclusively on simulating the behavior of ants.

It was a program called Tracker, designed in the mideighties by two UCLA professors, David Jefferson and Chuck Taylor. (Jefferson was in the computer science department, while Taylor was a biologist.) "I got the idea from reading Richard Dawkins's first book, *The Selfish Gene,*" Jefferson says today. "That book really transformed me.

He makes the point that in order to watch Darwinian evolution in action, all you need are objects that are capable of reproducing themselves, and reproducing themselves imperfectly, and having some sort of resource limitation so that there's competition. And nothing else matters—it's a very tiny, abstract axiom that is required to make evolution work. And so it occurred to me that programs have those properties—programs can reproduce themselves. Except that they usually reproduce themselves *exactly*. But I recognized that if there was a way to have them reproduce imperfectly, and if you had not just one program but a whole population of them, then you could simulate evolution with the software instead of organisms."

After a few small-scale experiments, Jefferson and Taylor decided to simulate the behavior of ants learning to follow a pheromone trail. "Ants were on my mind—I was looking for simple creatures, and E. O. Wilson's opus on ants had just come out," Jefferson explains. "What we were really looking for was a simple task that simple creatures perform where it wasn't obvious how to make a program do it. Somehow we came up with the idea of following a trail—and not just a clean trail, a noisy trail, a broken trail." The two scientists created a virtual grid of squares, drawing a meandering path of eighty-two squares across it. Their goal was to evolve a simple program, a virtual ant, that could navigate the length of the path in a finite amount of time, using only limited information about the path's twists and turns. At each cycle, an ant had the option of "sniffing" the square ahead of him, advancing forward one square, or turning right or left ninety degrees. Jefferson and Taylor gave their ants one hundred cycles to navigate the path; once an ant used up his hundred cycles, the software tallied up the number of squares on the trail he had successfully landed on and gave him a score. An ant that lost his way after square one would be graded 1; an ant that successfully completed the trail before the hundred cycles were up would get a perfect score, 82.

The scoring system allowed Jefferson and Taylor to create fitness criteria that determined which ants were allowed to reproduce. Tracker began by simulating sixteen thousand ants—one for each of the Connection Machine's processors—with sixteen thousand more or less random strategies for trail navigation. One ant might begin with the strategy of marching straight across the grid; another by switching back and forth between ninety-degree rotations and sniffings; another following more baroque rules. The great preponderance of these strategies would be complete disasters, but a few would allow a stumble across a larger portion of the trail. Those more successful ants would be allowed to mate and reproduce, creating a new generation of sixteen thousand ants ready to tackle the trail.

The path—dubbed the John Muir Trail after the famous environmentalist—began with a relatively straightforward section, with a handful of right-hand turns and longer straight sections, then steadily grew more complicated. Jefferson says now that he designed it that way because he was worried that early generations would be so incompetent that a more challenging path would utterly confound them. "You have to remember that we had no idea when we started this experiment whether sixteen thousand was anywhere near a large enough population to seek Darwinian evolution," he explains. "And I didn't know if it was going to take ten generations, or one hundred generations, or ten thousand generations. There was no theory to guide us quantitatively about either the size of the population in space or the length of the experiment in time."

Running through one hundred generations took about two hours; Jefferson and Taylor rigged the system to give them real-time updates on the most talented ants of each generation. Like a stock ticker, the Connection Machine would spit out an updated number at the end of each generation: if the best trail-follower of one generation managed to hit fifteen squares in a hundred cycles, the Connection Machine would report that 15 was the current record and then move on to the next generation. After a few false starts because of bugs, Jefferson and Taylor got the Tracker system to work—and the results exceeded even their most optimistic expectations.

"To our wonderment and utter joy," Jefferson recalls, "it succeeded the first time. We were sitting there watching these numbers come in: one generation would produce twenty-five, then twenty-five, and then it would be twenty-seven, and then thirty. Eventually we saw a perfect score, after only about a hundred generations. It was mind-blowing." The software had evolved an entire population of expert trail-followers, despite the fact that Jefferson and Taylor had endowed their first generation of ants with no skills whatsoever. Rather than engineer a solution to the trail-following problem, the two UCLA professors had evolved a solution; they had created a random pool of possible programs, then built a feedback mechanism that allowed more successful programs to emerge. In fact, the evolved programs were so successful that they'd developed solutions custom-tailored to their environments. When Jefferson and Taylor "dissected" one of the final champion ants to see what trail-following strategies he had developed, they discovered that the software had evolved a preference for making right-hand turns, in response to the three initial right turns that Jefferson had built into the John Muir Trail. It was like watching an organism living in water evolving gills: even in the crude, abstract grid of Tracker, the virtual ants evolved a strategy for survival that was uniquely adapted to their environment.

By any measure, Tracker was a genuine breakthrough. Finally the tools of modern computing had advanced to the point where you could simulate emergent intelligence, watch it unfold on the screen in real time, as Turing and Selfridge and Shannon had dreamed of doing years before. And it was only fitting that Jefferson and Taylor had chosen to simulate precisely the organism most celebrated for its emergent behavior: the ant. They began, of course, with the most elemental form of ant intelligence—sniffing for pheromone trails—but the possibilities suggested by the success of Tracker were endless. The tools of emergent software had been harnessed to model and understand the evolution of emergent intelligence in real-world organisms. In fact, watching those virtual ants evolve on the computer screen, learning and adapting to their environments on their own, you couldn't help wonder if the division between the real and the virtual was becoming increasingly hazy.

In Mitch Resnick's computer simulation of slime mold behavior, there are two key variables, two elements that you can alter in your interaction with the simulation. The first is the number of slime mold cells in the system; the second is the physical and temporal length of the pheromone trail left behind by each cell as it crawls across the screen. (You can have long trails that take minutes to evaporate, or short ones that disappear within seconds.) Because slime mold cells collectively decide to aggregate based on their encounters with pheromone trails, altering these two variables can have a massive impact on the simulated behavior of the system. Keep the trails short and the cells few, and the slime molds will steadfastly refuse to come together. The screen will look like a busy galaxy of shooting stars, with no larger shapes emerging. But turn up the duration of the trails, and the number of agents, and at a certain clearly defined point, a cluster of cells will suddenly form. The system has entered a phase transition, moving from one discrete state to another, based on the "organized complexity" of the slime mold cells. This is not gradual, but sudden, as though a switch had been flipped. But there are no switch-flippers, no pacemakers—just a swarm of isolated cells colliding with one another, and leaving behind their pheromone footprints.

Histories of intellectual development—the origin and spread of new ideas—usually come in two types of packages: either the "great man" theory, where a single genius has a eureka moment in the lab or the library and the world is immediately transformed; or the "paradigm shift" theory, where the occupants of the halls of science awake to find an entirely new floor has been built on top of them, and within a few years, everyone is working out of the new offices. Both theories are inadequate: the great-man story ignores the distributed, communal effort that goes into any important intellectual advance, and the paradigm-shift model has a hard time explaining how the new floor actually gets built. I suspect Mitch Resnick's slime mold simulation may be a better metaphor for the way idea revolutions come about: think of those slime mold cells as investigators in the field; think of those trails as a kind of institutional memory. With only a few minds exploring a given problem, the cells remain disconnected, meandering across the screen as isolated units, each pursuing its own desultory course. With pheromone trails that evaporate quickly, the cells leave no trace of their progress—like an essay published in a journal that sits unread on a library shelf for years. But plug more minds into the system and give their work a longer, more durable trail—by publishing their ideas in best-selling books, or founding research centers to explore those ideas—and before long the system arrives at a phase transition: isolated hunches and private obsessions coalesce into a new way of looking at the world, shared by thousands of individuals.

This is exactly what happened with the bottom-up mind-set over the past three decades. After years of disconnected investigations, the varied labors of Turing, Shannon, Wiener, Selfridge, Weaver, Jacobs, Holland, and Prigogine had started a revolution in the way we thought about the world and its systems. By the time Jefferson and Taylor started tinkering with their virtual ants in the mideighties, the trails of intellectual inquiry had grown long and interconnected enough to create a higher-level order. (Call it the emergence of emergence.) A field of research that had been characterized by a handful of early-stage investigations blossomed overnight into a densely populated and diverse landscape, transforming dozens of existing disciplines and inventing a handful of new ones. In 1969, Marvin Minsky and Seymour Papert published "Perceptrons," which built on Selfridge's Pandemonium device for distributed pattern recognition, leading the way for Minsky's bottom-up Society of Mind theory developed over the following decade. In 1972, a Rockefeller University professor named Gerald Edelman won the Nobel prize for his work decoding the language of antibody molecules, leading the way for an understanding of the immune system as a self-learning pattern-recognition device. Prigogine's Nobel followed five years later. At the end of the decade, Douglas Hofstadter published *Gödel, Escher, Bach,* linking artificial intelligence, pattern recognition, ant colonies, and "The Goldberg Variations." Despite its arcane subject matter and convoluted rhetorical structure, the book became a best-seller and won the Pulitzer prize for nonfiction.

By the mideighties, the revolution was in full swing. The Santa Fe Institute was founded in 1984; James Gleick's book *Chaos* arrived three years later to worldwide adulation, quickly followed by two popular-science books each called *Complexity.* Artificial-life studies flourished, partially thanks to the success of software programs like Tracker. In the humanities, critical theorists such as Manuel De Landa started dabbling with the conceptual tools of self-organization, abandoning the then-trendy paradigm of poststructuralism or cultural studies. The phase transition was complete; Warren Weaver's call for the study of organized complexity had been vigorously answered. Warren Weavers's "middle region" had at last been occupied by the scientific vanguard.

We are now living through the third phase of that revolution. You can date it back to the day in the early nineties when Will Wright released a program called SimCity, which would go on to become one of the best-selling video-game franchises of all time. SimCity would also inaugurate a new phase in the developing story of self-organizing: emergent behavior was no longer purely an object of study, something to interpret and model in the lab. It was also something you could *build*, something you could interact with, and something you could sell. While SimCity came out of the developing web of the bottom-up worldview, it suggested a whole new opening: SimCity was a work of culture, not science. It aimed to entertain, not explain.

Ten years after Wright's release of SimCity, the world now abounds with these man-made systems: online stores use them to recognize our cultural tastes; artists use them to create a new kind of adaptive cultural form; Web sites use them to regulate their online communities; marketers use them to detect demographic patterns in the general public. The video-game industry itself has exploded in size, surpassing Hollywood in terms of raw sales numbers—with many of the best-selling titles relying on the powers of digital self-organization. And with that popular success has come a subtle, but significant, trickle-down effect: we are starting to *think* using the conceptual tools of bottom-up systems. Just like the clock maker metaphors of the Enlightenment, or the dialectical logic of the nineteenth century, the emergent worldview belongs to this moment in time, shaping our thought habits and coloring our perception of the world. As our everyday life becomes increasingly populated by artificial emergence, we will find ourselves relying more and more on the logic of these systems—both in corporate America, where "bottom-up intelligence" has started to replace "quality management" as the mantra of the day, and in the radical, antiglobalization protest movements, who explicitly model their pacemakerless, distributed organizations after ant colonies and slime molds. Former vice president Al Gore is himself a devotee of complexity theory and can talk for hours about what the bottom-up paradigm could mean for reinventing government. Almost two centuries after Engels wrestled with the haunting of Manchester's city streets, and fifty years after Turing puzzled over the mysteries of a flower's bloom, the circle is finally complete. Our minds may be wired to look for pacemakers, but we are steadily learning how to think from the bottom up.