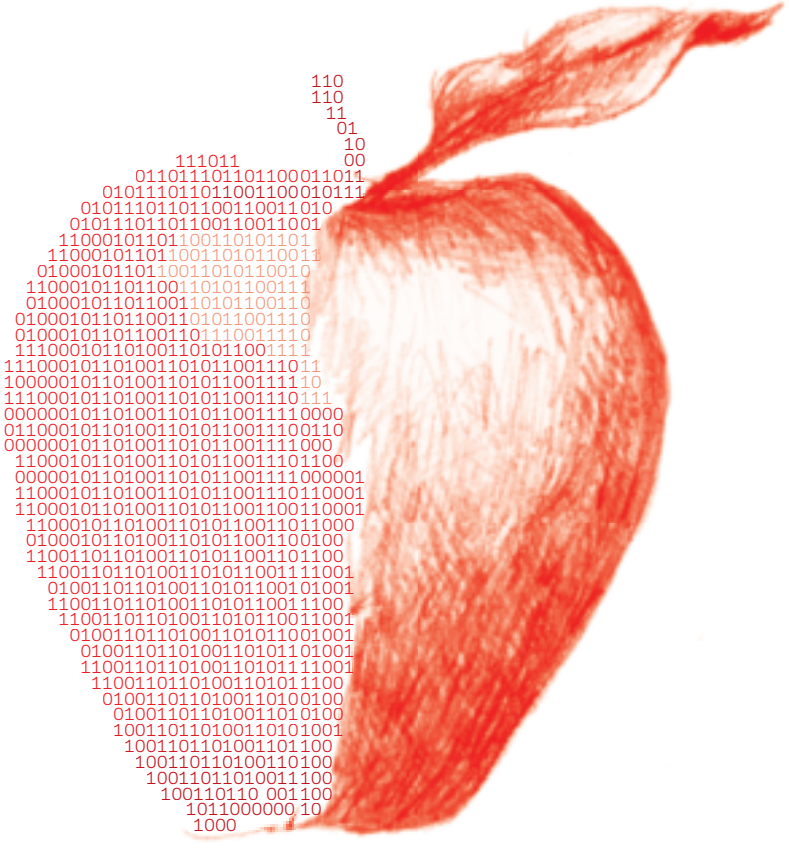


THE SCIENCE AND ART OF BUSINESS

by **Roger Martin**

For competitive advantage in the 21st century, firms must be able to access and balance both the science of business *and* the art of business.

I WAS INSPIRED TO EXAMINE the inner workings of the science and art of business by two occurrences in April of 2007 that framed a tension that I felt needed to be explored. The first was a speech by **Martin Baily**, former chair of **President Clinton**'s **Council of Economic Advisors** and at that time senior adviser to the **McKinsey Global Institute** (MGI), which is the think tank of what most would agree is the world's premier management consulting firm. The speech was on MGI's annual Ten Trends to Watch, and Trend #9 was, "The rise of scientific management and its triumph over gut instinct and intuition." Baily waxed eloquent about what an advance this represented for the world of business.

Intrigued, the following day I went to the MGI website to see exactly what was meant by Trend #9. Did Baily overstate the case in his speech? Did I hear him wrong? Neither: Baily was tame in comparison to the actual article by McKinsey's worldwide managing director **Ian Davis** titled "Ten Trends to Watch in 2006" and advertised as: "Macroeconomic factors, environmental and social issues, and business and industry developments [that] will profoundly shape the corporate landscape in the coming years."

> **Trend #9: Management will go from art to science.** Bigger, more complex companies demand new tools to run and manage them. Indeed, improved technology and statistical-control tools have given rise to new management approaches that make even mega-institutions viable.
>
> Long gone is the day of the 'gut instinct' management style. Today's business leaders are adopting algorithmic decision-making techniques and using highly-sophisticated software to run their organizations. Scientific management is moving from a skill that creates competitive advantage to an ante that gives companies the right to play the game.

Gut instinct and judgment: into the ashbin of history! This was heady stuff: leaders reducing the whole of business to sets of algorithms and suites of sophisticated software. "O brave new world, that has such people in't!" to borrow, as did **Aldous Huxley**, from **Shakespeare**.

The second occurrence happened two weeks after the dinner when a major article on the growing interest in design on the part of business, for which I was interviewed, appeared in *The Globe and Mail*, Canada's national newspaper. It was written not by a business writer but rather by an arts columnist, **Sarah Milroy**, and it began with the following snippet of a poem from **William Blake**:

> I must create a system
> Or be enslav'd by another man's
> I will not reason and compare:
> My business is to create.

It struck me when I read this stanza that there could be no greater contrast between fellow Brits Ian Davis and William Blake writing 200 years apart: running algorithms and software versus the act of creation.

While there is certainly an important kernel of truth in MGI's Trend #9, it is predicated on an overly-narrow view of value creation in the business world, a static view focused on only part of the overall knowledge domain in which businesses operate. That narrow view generates a limited definition of value-added and expertise: the running of algorithms and sophisticated software. Instead, I will argue that for competitive advantage in the 21st century, firms must be able to access and balance both the science of business *and* the art of business.

Only firms that master the creative design of Blake's 'systems' will prosper in the long run. A side-benefit for society is that business leaders who stay continuously focused on the creative design of systems will produce the greatest forward march of knowledge, which both benefits their firms and the world at large. On the other hand, managers who focus on running existing algorithms and software exploit their environment as they find it, but create little knowledge that moves the world forward.

I describe the thinking process required for Blake's creative design of systems as 'design thinking' . To understand the profound difference between analytical thinking and design thinking, we need to step back and understand the process by which knowledge advances.

The Advancement of Knowledge

Over the course of time, phenomena enter our collective consciousness as *mysteries* – things that we observe and that intrigue us, but that we don't yet understand. For instance, the mystery of gravity once confounded our forefathers: when they looked around at the world, they saw that most objects, apples famously, seemed to fall to the ground quickly; but others didn't, such as birds; and some fell but seemed to take forever, like leaves. In art, there was the long struggle to understand how to represent on a two-dimensional surface what we saw in front of us in three dimensions. Music continues to be a mystery that confounds most of us: what patterns of notes and sounds are enjoyable and make listeners feel contented or inspired?

We start out with these mysteries, and at some point, because they intrigue us, we put enough thought into them to produce a first-level understanding of the question at hand. We develop *heuristics* – ways of understanding the general principles of what were heretofore mysteries. Heuristics are rules of thumb or guidelines for solving a mystery by way of organized exploration of the possibilities.

So why do things fall down? We develop a notion of a universal force called 'gravity' that tends to pull things down. In art, we develop a notion called 'perspective' that guides our efforts to create renderings that appear to the eye to have three dimensions rather than two. What kind of music do people like to listen to? We learn about chords, and then create song types like ballads, or folk songs, or the blues. By following a set of guidelines, we increase the likelihood of creating something that people will enjoy listening to.

The application of heuristics doesn't guarantee success: it simply increases the probability and/or speed of getting to a successful outcome. Heuristics represent an incomplete-yet-distinctly-advanced understanding of what was previously a mystery. In any given field, some people remain stuck in the world of mystery, while others master its heuristics. The difference between them is the difference between one-hit-wonder **Don McLean**, composer of "American Pie", and **Bruce Springsteen**, whose eight number-one albums have sold 120 million copies worldwide. For McLean, the mystery remained just that: he came up with a single inspiration that created one random event – for several decades the biggest-selling pop song of all time. Yet he failed to produce another hit of any consequence in his entire career. In contrast, Springsteen developed a heuristic – a way of understanding the world and the people in it – that enables him to write songs that have great meaning to people. His mastery of heuristics has allowed him to generate a steady stream of hits over a 30-plus-year period.

In due course, increased understanding can – though in many cases it never does – produce an *algorithm*: a logical, arithmetic or computational procedure that, if correctly applied, ensures the solution of the problem. With gravity, great scientists like **Sir Isaac Newton** studied and experimented long and hard enough to create precise rules for determining how fast an object will fall under any circumstance. In the late 1970s, musical innovators like British techno-music guru **Brian Eno** experimented with the human heartbeat and determined that songs with a synthesized heartbeat as their rhythm track are instinctively enjoyed by listeners, no matter what score you add on top of the heartbeat. That enabled bands whose songs he produced to experience consistent success. The end result of such algorithms is not always positive, of course: this discovery also led to electro-pop and eventually to sham bands like **Milli Vanilli**, who lip-synched recorded music onstage until caught in the act by an unsuspecting audience. And in art, we eventually got paint-by-numbers – the ultimate algorithm.

In the modern era, a fourth important step has been added to the sequence of *mystery* to *heuristic* to *algorithm*: eventually, some algorithms can be coded into software. This means reducing the algorithm – the strict set of rules – into a series of 0's and 1's – *binary code* – that enables a computer to produce the desired result. In the case of gravity, the fact that we had an algorithm for 'how things fall' meant that **Honeywell** engineers could program aircraft with autopilot, enabling a plane to 'fall' from the sky in the controlled fashion that we want it to, so that it lands in exactly the right spot. At the coding level, there is no longer any judgment involved: the plane lands on the basis of computer instructions that are nothing but a series of 1's and 0's, because our understanding of gravity has moved from *mystery* to *heuristic* to *algorithm* to *code*.

The Creation of Value in Business

With the sequence of *mystery* to *heuristic* to *algorithm* to *code* in mind, we can consider the question of how value can be created in business. The answer is, in two fundamental ways: first, value can be created by operating a business within a single knowledge category; i.e. either by running a heuristic, an algorithm, or code; and second, value can be created by instigating a progression along the sequence of knowledge from mystery to heuristic, heuristic to algorithm or algorithm to code.

To illustrate the two types of value creation, let's explore **McDonald's Corporation**. In 1955, at a time of the emergence of the freeways and beach culture in Southern California – a unique and leading-edge environment within America – the McDonald brothers stared into the face of a mystery: how and what do Californians want to eat in this emerging cultural environment? After thinking about it for sufficiently long and experimenting based on their ideas, they created a format for answering that – a heuristic – which was the quick-service restaurant with a limited menu for fast turnaround of the food ordered and time-saving devices like the three-at-a-time milkshake maker.

This heuristic created sizable, though by no means enormous, value for the McDonald brothers: they went on to open additional outlets, and by the time an investor named **Ray Croc** approached them to buy the chain, they had four very successful outlets and

As one moves from mystery toward code, one does so by dropping things out of the equation – by simplifying the complexities of the world.

were making a handsome return. Croc bought their fledgling chain and saw that he could drive the McDonald brothers' heuristic to an algorithm. He figured out *exactly* how to cook a hamburger, *exactly* how to hire people, *exactly* where to set up restaurants, *exactly* how to manage stores, and *exactly* how to franchise them. Under Kroc, nothing was left to chance in the McDonald's kitchen: every hamburger came out of a stamping machine weighing exactly 1.6 ounces, its thickness measured to the thousandth of an inch, and the cooking process stopped automatically after 38 seconds, when the burgers reached an internal temperature of exactly 155 degrees.

Kroc created value by driving a heuristic to an algorithm, and then additional value by running that algorithm and building McDonald's into a global firm of leading size and scope. Thus he demonstrates the two forms of value creation: progression along the sequence of understanding – in this case from heuristic to algorithm – and running one stage – the algorithm.

Achieving the Required Balance

Today, both running an operation efficiently within one knowledge stage and advancing it across knowledge stages are critical to an organization's success. The great management theorist **James March** argues that every organization needs a balance of *exploration* and *exploitation*: exclusive exploration – i.e. seeking to advance knowledge – causes an organization to expire in relatively short order because it produces few resources with which to continue to engage in exploration. On the other hand, exclusive exploitation – i.e. seeking to maximize the payoff from existing knowledge – causes an organization to expire in due course because it can't keep exploiting the same piece of knowledge forever.

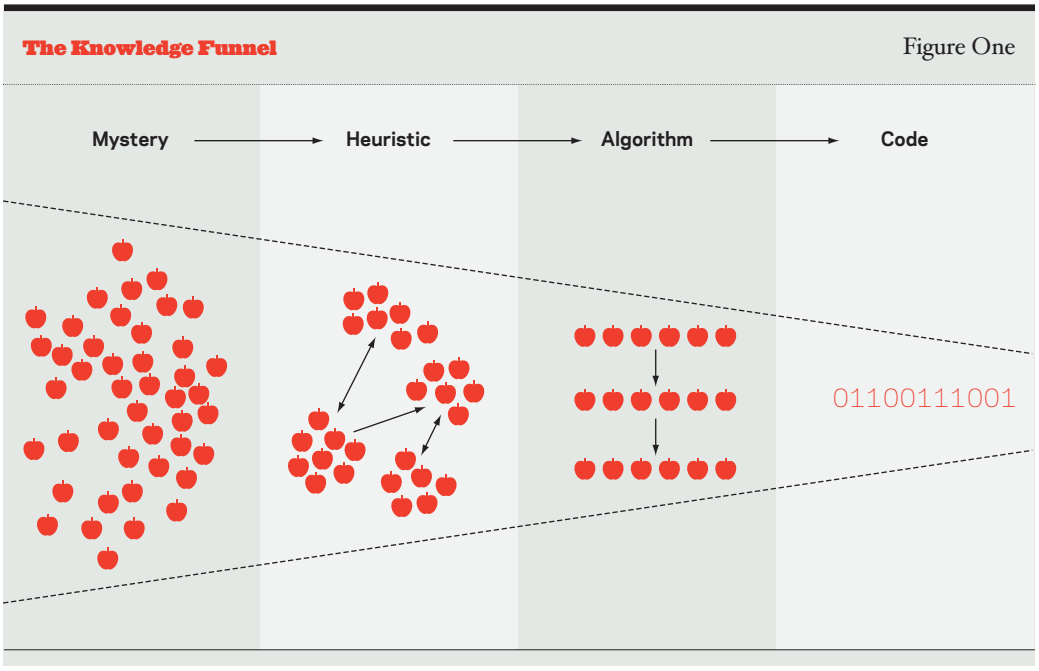
McDonald's provides a prime example of the tricky balance that is required. From the 1960s to the 1980s, it succeeded in exploiting the quick-service restaurant algorithm based on burgers, fries and milkshakes. What it failed to do was stare into the mystery of what consumers in the 1990s wanted by way of fast food. It turned out that many of them still wanted the quickness that McDonald's provided, but they also wanted something that they perceived to be healthier than pressure-cooked beef and deep-fried potatoes. Other chains, from **Taco Bell** to **Subway**, explored and found out what those consumers wanted and drove McDonald's into a tailspin. Only when it began to explore new approaches to satisfying the new consumer did it start to bounce back from its deepest performance woes; but alas, it was far too late to forestall the arrival and maturation of a large-scale challenging competitor like Subway.

The McDonald's story illustrates important elements of the dynamics of the march of knowledge from mystery to heuristic to algorithm to code. As one moves from mystery toward code, one does so by dropping things out of the equation; by simplifying the complexities of the world.

Consider the first step: to stare into a mystery and develop a heuristic, one has to come up with an explanatory model that answers the question, 'why am I seeing what I am seeing?' Any answer is by necessity a simplification, because otherwise one would merely be describing what one sees in its entirety, which provides no gain in understanding. The gain in understanding comes from picking out features that are salient and building a causal understanding out of them – i.e. "I think that Californians would like a quick-service hamburger joint." The heuristic doesn't attempt to understand everything about the ethnographic mysteries of this new Californian beach culture, but rather focuses on one specific aspect of it – desired out-of-home eating experience.

To move to the next step and create an algorithm, still more vast tracts of possibility must be shunted aside: hamburgers could be charbroiled or pressure-cooked; the menu could be broad or narrow; restaurants could be small or large, etc. Essentially, one answer along innumerable dimensions had to be plucked to provide the formula that defined the McDonald's algorithm. Along the way, judgment was removed, possibilities were removed and variety was removed. That is anything but bad – it is absolutely necessary to make the move from heuristic to algorithm.

While McDonald's didn't take the progression to the fourth and final step, even further narrowing would have been required to take its operations to the point of software code. While an algorithm still requires some judgment – for example, in measuring the variables



that are used in the algorithm – binary code removes all judgment. Even if the circumstance calls for some other nuance, the cold calculating lines of software code will not take that into consideration.

The Knowledge Funnel

Advancing from mystery to code does not come without costs: by its very nature, it involves a significant narrowing of the problem space. The entire mystery may be the object of contemplation and exploration, but knowledge only advances as a narrower and narrower slice of the whole is captured and worked on. In essence, the advance of knowledge can be seen as a 'funnel' moving from broad to narrow, as shown in **Figure One**.

The key implication of the Knowledge Funnel is that as a business moves toward algorithm and code, it is leaving out more and more of the big picture. And if the business in question focuses entirely on running its algorithm and/or code, it will be entirely vulnerable to other organizations that continue to stare into aspects of the mystery that the algorithm-runner has left out. This will enable the other organizations to create alternative- and potentially-superior heuristics, algorithms and code – because there will always be multiple paths out of any mystery. While the algorithm-runner is standing still, using analytical thinking to exploit the value of its necessarily-narrowed insights, others are likely advancing knowledge to use against it, whether it be **Taco Bell** figuring out how to get mainstream Americans to eat Mexican food, **Subway** figuring out how to make health an important component of a quick service offering, or **Whole Foods** convincing customers to substitute healthy prepared foods purchased at a grocer for the offerings of quick-service restaurants.

In closing

While there is a powerful grain of truth to the notion that managers should exploit their algorithms and software to the maximum extent possible, doing so by way of eliminating judgment (even if it is called by its pejorative name, 'gut instinct') is dangerous advice. Even if done brilliantly well, all that this guarantees is temporary dominance of a necessarily-narrowed domain. And it almost guarantees that the business will be superseded by a competitor that uses judgment to explore mysteries to which the algorithm-runner has consciously or unconsciously averted his eyes. In the end, it is all about choosing *and* over *or*: combining the science *and* the art of business is the path to creative solutions. ***R***

Roger Martin is dean, professor of Strategic Management, Premier's Chair in Competitiveness and Productivity and director of the AIC Institute for Corporate Citizenship at the Rotman School of Management. He is the author of *The Opposable Mind: How Successful Leaders Win Through Integrative Thinking* (Harvard Business Press, 2007), and the forthcoming *The Design of Business* (Harvard Business Press, 2009).