CHAPTER

Learning From Experience and Performing Learned Actions are Easy; Novel Actions, Problem-Solving, and Calculation are Hard

10

As we saw in Chapter 9's comparison of recognition and recall, the human brain is good at some tasks and not so good at others. In this chapter, we compare several additional functions of the brain to show those it is good at and those it is bad at as well as to see how to design computer systems to support human activity. But first, a bit more about the brain and mind.

WE HAVE TWO BRAINS

The human brain has many component parts. Each part consists of neurons and supporting tissue: glial (glue) cells, blood vessels, etc. The parts interconnect and interact to a great degree, but each plays a role—in some cases multiple roles—in making us a fully functioning human being. Brain researchers organize the brain's components differently depending on whether they are interested in the brain's anatomy (physical structure) or physiology (function), but for our purposes we will consider the human brain as consisting of two main groups of components: "old brain" components and "new brain" components.

The old brain

The old brain is called "old" because its components evolved millions of years ago. Most were present in ancestors of today's vertebrates: amphibians, reptiles, birds, and mammals.[1] The components that together comprise the old brain are:

- **Brain stem**: It starts where the spinal cord enters the skull and consists of the medulla, pons, midbrain, and reticular formation. It regulates autonomic behavior:

[1] Invertebrate animals such as insects, spiders, worms, and molluscs don't have brains in the usual sense of the word.

Designing with the Mind in Mind. https://doi.org/10.1016/B978-0-12-818202-4.00010-6

breathing, heart rate, and automatic physical movements like standing and walking. The brain stem also regulates sensory input coming through the spinal cord from the body and relays it to other parts of the brain.

- **Hypothalamus**: Its name means "below the thalamus." It regulates hunger, thirst, sexual behavior, and other functions and links to the pituitary gland (part of the endocrine system).
- **Thalamus**: Positioned at the top of the brain stem, it regulates two-way communication between the new brain and the brain stem, including blocking sensory input while we sleep.
- **Cerebellum**: Its name means "little brain" because it consists of two wrinkled balls behind the brain stem. Its main function is to coordinate voluntary movement, but it also is involved in emotions, distinguishing sounds and textures, and learning (Bower and Parsons, 2003).
- **Amygdala**: It is two almond-shaped neuron clusters near the top of the brain stem. It is heavily involved in our perception and memory of events that induce aggression and fear and helps regulate our reactions to such events.
- **Hippocampus**: This is two horn-shaped bodies behind the amygdala that wrap around the thalamus. Researchers are still learning about the hippocampus, but its primary function seems to be regulating the storage of our experiences into long-term memory.

Considered as a whole, the old brain's main role seems to be to govern our automatic behavior—the things we do without conscious thought. However, that is an oversimplification since (1) many parts of the old brain connect to the new brain and help it do what it does and (2) parts of the new brain also govern automatic behavior. Researchers are constantly discovering new functions of each of the old brain's components.[2]

The new brain

It is called "new" because it appeared relatively recently in the evolution of animal species. Its formal name is the *cerebrum*. All modern vertebrates—animals with backbones—have it, but it is nearly unnoticeable in fish, lizards, and birds.[3] Only in mammals is it prominent. The components of the new brain (cerebrum) are:

- **Cerebral cortex**: This is a layer of neurons covering the cerebrum about 2–4 mm thick. In lower mammals, such as rats, it is smooth, but in higher animals, such as apes, cetaceans, and humans, it is convoluted, allowing a large surface area

[2]Some anatomy textbooks classify the cerebellum, amygdala, and hippocampus as part of the new brain.

[3]Fish, lizards, and birds were once thought not to have new-brain structures, but now it is recognized that they do (Dubuc, 2012).

to fit inside the skull. The cortex is divided down the middle into left and right hemispheres, with each hemisphere controlling the opposite side of the body. Each hemisphere consists of four lobes. Starting from the rear of the brain, the ***occipital*** lobe's main role is to process visual input. The ***parietal*** lobe (top rear) processes sensations of touch, temperature, and taste and integrates them with inputs from vision and hearing. The ***temporal*** lobe (sides) processes auditory and olfactory inputs, integrates vision with language (e.g., connects objects and faces with their names), and helps form long-term memories. The ***frontal*** lobe initiates and controls voluntary movements (while the cerebellum coordinates them), processes short-term memories, and is involved in storing long-term *episodic* memories, such as memories of your last birthday party or of a recent trip to a museum, and long-term *semantic* memories, such as the meaning of the word "government." A small area of the frontal lobe—the *prefrontal* cortex—seems to be involved with conscious thought and decisions, inhibiting undesirable behavior, and creating a sense of *self* versus *others.*

- **Corpus callosum**: Its name means "hard body." It is a curved bar of neurons connecting the left and right hemispheres of the cerebrum. That seems to be its main role—to be the main communications channel between the hemispheres.
- **Basal ganglia**: This is a pair of tadpole-shaped neuron clusters that loop around the corpus callosum and thalamus, one in each hemisphere. Functionally the basal ganglia seem to be involved in the formation of long-term *procedural* memories, such as how to tie a shoe, ride a bike, play a tune, or drag an app's icon to the trash.

In primates, including humans, the new brain accounts for most of the brain's size, weight, and energy consumption. Overall, human brains consume about 20% of the calories we eat, and most of those are used by our new brain (Eagleman, 2015).

WE HAVE TWO MINDS

Cognitive psychologists view the human mind as consisting of two distinct "minds": an unconscious, automatic mind operating largely in the old brain and most of the new brain, and a conscious, monitored mind operating mainly in the frontal cortex of the new brain, specifically in the very front—an area called the *prefrontal cortex*. Psychologists often call the unconscious, automatic mind *system one* because it evolved first and is the main controller of our perception and behavior. They refer to the conscious, monitored mind as *system two* because (1) it came into existence very recently in evolutionary time frames—perhaps only within the last few million years—and (2) it usually takes a back seat in controlling human perception and behavior (Kahneman, 2011).

One noteworthy fact about our conscious, rational, monitored mind (system two) is that it is *us*—it is where our consciousness and self-awareness are. Of course, it thinks it is in charge of our behavior. It believes it runs the show because it is the only one of the two minds that *has* consciousness and a sense of *self*. But in fact system two is *rarely* in charge (Kahneman, 2011; Eagleman, 2012, 2015).

System one (the unconscious, automatic mind) operates quickly compared with system two—10 to 100 times as fast—but it does so by operating based on intuition, guesses, and shortcuts, which makes everything it does an approximation. For example, consider this math problem (adapted from Kahneman, 2011):

A baseball and a bat together cost $110. The bat costs $100 more than the ball. How much does the ball cost?

Most likely, your system one instantly gave you (i.e., your system two) the answer: $10. Perhaps your system two accepted that answer. Or maybe after a moment of thought, it rejected it. If the ball costs $10 and the bat costs $100 more (i.e., $110), then their combined cost would be $120. But as we said at the start, the two add up to $110, so the ball cannot cost $10. What is the correct answer? You have to engage your system two to figure it out.[4]

System one is also easily biased. The perceptual biases described and illustrated in Chapter 1 are biases of system one. Look at Fig. 10.1. System one sees the dogs getting

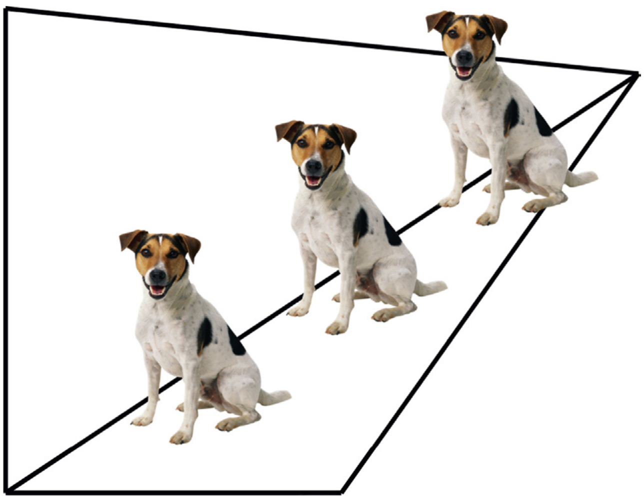


FIGURE 10.1

The three dogs are the same size, but the converging lines bias system one to see them as arranged in three dimensions and successively larger.

[4]The correct answer for the cost of the ball is $5.

larger as you go upward and to the right (i.e., toward the back in system one's view), but in fact they are the same size. Even after your system two *knows* this, it is hard to override system one.

System one has several other characteristics:

- When it encounters a problem it can't solve, it substitutes an easier problem and solves that. For example, accurately answering the question "Is asparagus a popular vegetable?" requires engaging system two and either conducting a survey or looking up and reading survey results, which takes time and effort. System one just tosses that question out and quickly answers the question "Do I like asparagus?"
- It bases judgments only on what it perceives; it doesn't care that other important (potentially conflicting) information might exist. If such data is not present, it doesn't exist.
- It filters perceptions based on goals and beliefs given to it by system two; information that doesn't match is filtered out before reaching system two.

We all have both system one and system two,[5] but system two is often lazy: it accepts the quick estimates and judgments of system one even though they are often inaccurate. Why? Because the perceptions and judgments of system one come quickly and are usually good enough to allow us to get by in most situations. Also, operating system two requires conscious will and mental effort, while system one is always running in the background and requires no conscious effort. Thirdly, system two is more expensive to operate; it consumes more energy (calories) than system one, so the brain avoids activating it unless the situation requires it. It prefers to rely on automatic processes (system one) whenever possible (Eagleman, 2015).

So why do we have a system two? Human behavior, like that of other animals, is run mainly by unconscious processes[6]—system one. But a fully automatic brain would be inflexible: it couldn't switch goals in midaction, quickly adjust its response to rapidly changing situations, or resolve conflicts when more than one automatic response could apply. To counter this, people—and perhaps a few other animals—also have a small conscious "CEO" process that can oversee and guide the operation of system one: set high-level goals, perform exact calculations, inhibit ill-advised actions, etc. Usually it isn't needed, so it "sleeps." But when needed, it wakes up and tries to assume control, sometimes successfully (Eagleman, 2015).

When is system two needed? When our goals require getting something not just *sort of* right but *exactly* right, when we are in situations system one does not recognize and therefore has no automatic response, when system one has multiple

---

[5]Kahneman (2011) points out that the distinction between system one and system two is not a scientific fact but rather an analogy concocted by psychologists to help explain the dual character of human cognition. The truth is more complex, but beyond the scope of this book.

[6]Brain researcher David Eagleman calls them "zombie" or "robotic" processes (Eagleman, 2012).

conflicting responses and no quick-and-dirty way to resolve them, or when system one is about to make us do something that would be detrimental to us in the long run.

Because system one is the primary controller of human perception and behavior, with system two intervening only as necessary, the human mind is not fully rational and conscious—it isn't even *mostly* rational and conscious. When we perceive something—an object or event—both minds react and contribute to our thought and behavior. Since system one reacts faster than system two, we sometimes act based on what it tells us before we (i.e., our system two) can reach a conscious decision or are even aware that action is required.

LEARNING FROM EXPERIENCE IS (USUALLY) EASY

People are pretty good at generalizing from specific experiences and observations to extract conclusions. We generalize constantly throughout our lives.

The neural basis of learning is not as well understood as that of recognition and recall (Liang et al., 2007). However, people learn from their experiences constantly and often automatically. Most people, given the necessary experience, easily learn such lessons as:

- Stay away from leopards.
- Don't eat bad-smelling food.
- Wait a day before replying to an email that makes you mad.
- Don't open attachments from unfamiliar senders.
- LinkedIn is useful, but Facebook is a waste of time (or vice versa, depending on your experience).
- Before asking Siri, Alexa, Cortana, or Google Assistant a question, think about how to phrase the question.

In fact, learning from experience and adjusting our behavior accordingly does not require our awareness that we are learning and adjusting. System one can do it alone without system two's involvement.

For example, imagine that you are in a casino playing two slot machines next to each other. Unbeknownst to you, the machines are rigged so one pays out slightly more often than the other. After you've played hundreds of times, you may be able to identify the "good" machine. That is system two talking. But if we measure your galvanic skin response (GSR, a measure of anxiety), we would see that after only a few dozen trials, your system one has already identified the "bad" machine: whenever you reach for it, your GSR jumps. System one might even make you start to avoid that slot machine unless overridden by your still-clueless system two.

However, our ability to learn from experience is limited in several ways. First, complex situations involving many variables or subject to a wide variety of forces are difficult for people to predict, learn from, and generalize about. For example:

- People who have lived in Denver, Colorado, for decades still have trouble predicting the weather there.
- Even experienced stock market investors cannot predict with certainty which stocks will rise or fall on a given day.
- Even highly experienced network engineers have trouble diagnosing some network bottlenecks or outages.

Second, experiences from our own lives or those of relatives and friends influence our conclusions more than experiences we read or hear about. For example, we may have read and seen reports, consumer reviews, and statistics indicating that the Toyota Prius is a great car, but if our Uncle Charlie had a bad experience with one, we will probably have a negative assessment of it. We do this because our system one considers family members as like us and therefore more trustworthy than data about thousands of anonymous car buyers, even though from a rational standpoint those statistics are more reliable (Weinschenk, 2009; Kahneman, 2011; Eagleman, 2012, 2015).

Third, when people make a mistake (see Chapter 15), they don't always learn the right lesson from it. By the time they realize they are in a bad situation, they may not remember their recent actions well enough to be able to connect their situation with the true cause or causes.

A fourth problem people have in learning from experience is that they often overgeneralize—that is, make generalizations based on incomplete data. As explained earlier, this is a characteristic of system one. For example, many people assume all crows are black because all the crows they have seen are black. In fact, some crows are not black (see Fig. 10.2).

However, it can be argued that overgeneralizing isn't a problem—it's a feature. It is rare that one can see all possible examples of something. For example, a person can never see all crows, but it may still be useful in daily life (although not in scientific research) to assume that the many crows one has seen are enough evidence to conclude that all crows are black. Overgeneralization, therefore, seems like a necessary adaptation for life in the real world. It is primarily when we overgeneralize in extreme ways—for example, on the basis of few or atypical examples—that we get ourselves into trouble.

For example, some people may avoid all apps from a certain software company because they previously had trouble with one or more of that company's apps. The company's other apps may be quite fine and the company may have improved its old apps, but some former customers will never know that because they have overgeneralized, ruling out ever buying anything from that company again. Furthermore, those customers can share their overgeneralization through online ratings, comments, and tweets, causing others to make the same overgeneralization. Overgeneralization can also work in a company's favor if a person likes one of its products and shares their enthusiasm via social media.

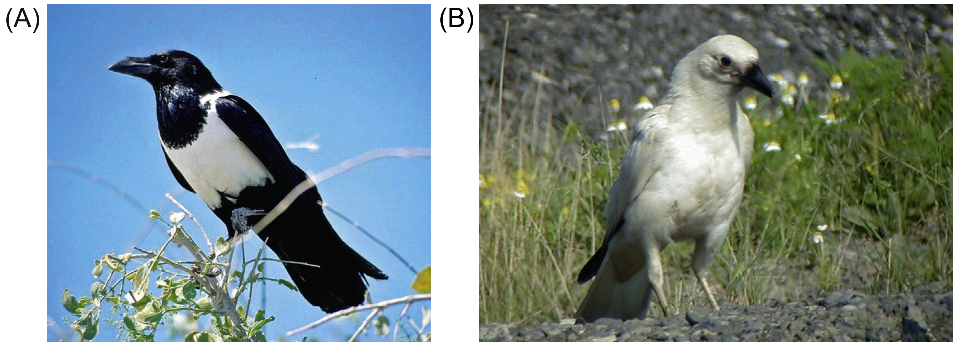


FIGURE 10.2

The common belief that all crows are black is false: (A) African pied crow and (B) white (nonalbino) crow, Ohio. ((A) Photograph by Thomas Schoch.)

The ability to learn from experience has a long evolutionary history. A creature does not need a cerebral cortex to be able to do it. Even insects, mollusks, and worms, without even an old brain—just a few neuron clusters—can learn from experience. However, only creatures with a cortex can learn from the experiences of *others*.[7] A cortex is certainly necessary to be *aware* that one has learned from experience, and only creatures with the largest prefrontal cortex (relative to body size)—possibly only humans—can *articulate* what they have learned from experience.

Bottom line: Even though there are limits on how well we learn from direct experience and the experiences of others, learning and generalizing from experience are relatively easy for the human mind.

PERFORMING LEARNED ACTIONS IS EASY

When we go somewhere we have been many times before or do something we have done many times before, we do it almost automatically without much conscious thought. The route, the routine, the recipe, the procedure, the action, has become semiautomatic or fully automatic. We are mainly using system one. Here are some examples:

- Riding a bicycle after many years of practice.
- Backing out of your driveway and driving to work for the 300th time.
- Playing a tune that you have played hundreds of times on a musical instrument.
- Scrolling a display on a touch screen by swiping your finger.

[7]Some birds can learn from watching other birds, so apparently even the tiny cortex birds have is enough.

- Entering a banking transaction into your online bank account app.
- Checking your email and replying to some of your messages.
- Reading and then deleting a text message from your longtime mobile phone.

In fact, "automatic" is how cognitive psychologists refer to routine, well-learned behavior (Schneider and Shiffrin, 1977). Researchers have determined that performing this type of action consumes few or no conscious cognitive resources—that is, it is not subject to the limits of attention and short-term memory described in Chapter 7. It also consumes relatively little energy compared with that of conscious cognition (Eagleman, 2015).

Automatic activities can even be done (by system one) in parallel with other activities. Thus, you can tap your foot while humming a familiar song while beating an egg, while still leaving your conscious mind (system two) "free" to keep an eye on your children or plan your upcoming vacation.

How does an activity become automatic? The same way you get to Carnegie Hall (as the old joke goes): practice, practice, practice. Practicing an activity "burns" it into system one so it becomes increasingly more automatic. It is ironic that people, as a species, usually consider conscious, intentional, planned activity the most exalted type of activity—proof of our "superiority" over other animals—yet we spend most of our lives trying to move most of what we do out of system two into system one so we can do it easily, without thinking.

PERFORMING NOVEL ACTIONS IS HARD

When a person first tries to drive a car, every part of the activity requires conscious attention (i.e., the engagement of system two). Am I in the right gear? Which foot do I use to press the accelerator pedal and brake pedal? How hard should I press on each pedal? Which pedal am I pressing now? Which way am I headed? How fast am I going? What is ahead of me, behind me, beside me? Where are the mirrors I should be checking? Is that my street coming up ahead? "Objects in mirror are closer than they appear"—what does that mean? And what is that light blinking on the dashboard?

When everything involved in driving a car is still conscious, keeping track of it all *far* exceeds our attention capacity—remember, that capacity is four items, plus or minus one (see Chapter 7). People who are still learning to drive often feel overwhelmed. That is why they often practice driving in parking lots, parks, rural areas, and quiet neighborhoods, where traffic is light—to reduce the number of things they have to attend to.

After much practice, slowly, one by one, the actions involved in driving a car become automatic; they are taken over by system one. They no longer compete for attention and recede from consciousness. We may not even be fully aware of doing them. For example, which foot do you use to push the accelerator pedal? To remember, you probably had to pump your feet briefly.

Similarly, when music teachers teach students to play a musical instrument, they don't make students monitor and control every aspect of their playing at once. That would overwhelm the students' attention capacity. Instead, teachers focus students' attention narrowly on one or two aspects of their playing: the correct notes, rhythm, tone, articulation, or tempo. Only after students learn to control some aspects of their playing without thinking about them do music teachers require their students to control more aspects simultaneously.

To demonstrate to yourself the difference in conscious attention required by well-learned (automatic) versus novel (controlled) tasks, try these:

- Recite the letters of the alphabet from A to M. Then recite the letters of the alphabet from M to A.
- Drive to work using your normal route. The next day, use a different, unfamiliar route.
- Hum the first measure of the song "Twinkle, Twinkle, Little Star." Then hum it backward.
- Enter your phone number using a standard 12-key telephone pad. Then enter your phone number using the number keys at the top of your computer keyboard.
- Type your full name on a computer keyboard. Then cross your hands over the keyboard and type your full name again.
- Schedule an online video call using your favorite video calling app. Then schedule one using a video calling app you have never used before.

Most real-world tasks have a mixture of automatic and controlled components. Driving to work along your usual route is largely automatic, allowing you to focus your attention on radio news or think about your evening dinner plans. Even your initial reaction to looming dangers, like a car suddenly merging into your lane or a child running out in front of your car, is automatic. But if you encounter an unexpected roadblock or traffic jam, your system two engages and your attention will be yanked back to the task of driving—specifically, figuring out an alternate route.

Similarly, if you check your email using your usual email program, the way you retrieve and view that email is well practiced and mostly automatic, and reading text is well practiced and automatic (see Chapter 6), but the *content* of any newly arrived email messages is new and therefore requires your conscious attention. If while on vacation you go into a hotel's business center and try to check your email using an unfamiliar computer, operating system, or email program, less of the task will be automatic, so it will require more conscious thought, take more time, and be more prone to error.

When people want to get something done—as opposed to challenging themselves mentally—they prefer to use methods that are automatic or at least semiautomatic to save time, mental effort, and energy and to reduce the chance of error. If you are in a hurry to pick up your child from school, you take your tried-and-true route even if your neighbor told you just yesterday about a faster route. Remember what the usability test subject said (see Chapter 8):

I'm in a hurry, so I'll do it the long way.

How can designers of interactive systems make the tasks that they support faster, easier, and less error-prone? By designing them so they can be handled by the automatic functions of system one or quickly become so. How does one do that? Chapter 11 describes some of the ways.

PROBLEM-SOLVING AND CALCULATION ARE HARD

Reptiles, amphibians, and most birds get along in their world quite well with just a system one.[8] Insects, spiders, and mollusks survive in their environments with even less. Animals without a cortex can learn from experience, but they can only learn minor adjustments to their behavior. Most of their behavior is stereotyped, repetitive, and predictable once we understand the demands of their environment (Simon, 1969). That may be just fine when their environment requires only the behaviors they already have automated.

But what if the environment throws a curve ball—it requires new behavior, and requires it *right now*? What if a creature faces a situation it has never encountered before and may never encounter again? In short, what if it is faced with a *problem*? In such cases, creatures without a cortex cannot cope.

Having a cerebral cortex (new brain) frees creatures from relying solely on instinctive, reactive, automatic, well-practiced behaviors. The cortex is where conscious reasoning happens in people (Monti et al., 2007). In current cognitive theory, it is largely where system two resides. Generally speaking, the larger a creature's cerebral cortex—specifically the prefrontal lobe of the cortex—relative to the rest of its brain, the greater its ability to interpret and analyze situations on the fly, plan or find strategies and procedures to cope with those situations, execute those strategies and procedures, and monitor their progress.

Expressed in computer jargon, having a system two gives us the ability to devise programs for ourselves on the fly and run them in an emulated, highly monitored mode rather than a compiled or native mode. That is essentially what we are doing when we are following a cooking recipe, playing bridge, calculating income taxes, following instructions in a software manual, or figuring out why no sound is coming out of the computer when we play a video.

[8]For example, salamanders choose a jar containing four fruit flies over one with two or three fruit flies (Sohn, 2003).

THE NEW BRAIN ALSO ACTS AS A BRAKE ON IMPULSIVE BEHAVIOR

The new brain—specifically the frontal cortex—also acts to inhibit reflexive and impulsive behavior coming from the old brain that could interfere with the execution of the new brain's carefully worked-out plans (Sapolsky, 2002). It keeps us from jumping up and getting off of a subway car when a smelly person boards, because after all, we do have to get to work on time. It keeps us sitting quietly in our seats at classical music concerts but lets us stand up and hoot and holler at rock concerts. It helps keep us out of fights (usually). It tries to stop us from buying that red sports car, because preserving our marriage is a higher goal than having the car. And whereas the old brain is tempted by the email that proposes a "BUSINESS OPPORTUNITY WORTH $12.5 MILLION DOLLARS," the new brain stops us from clicking, saying, "It's a spammer and a scammer; you know that, don't you?"

Although having a large new brain gives us the flexibility to deal with problems on short notice, that flexibility has a price. Learning from experience and performing well-learned actions are easy largely because they don't require constant awareness or focused attention and can occur in parallel. In contrast, controlled processing—including problem-solving and calculation—requires focused attention and constant conscious monitoring, and executes relatively slowly and serially (Schneider and Shiffrin, 1977). It strains the limits of our short-term memory because all the chunks of information needed to execute a given procedure compete with each other for scarce attention resources. It requires conscious mental effort, as you saw when you tried to recite the alphabet backward from M to A.

In computer jargon, the human mind has only one serial processor for emulation mode, or controlled execution of processes—system two. System two is severely limited in its temporary storage capacity, and its clock is between one and two orders of magnitude slower than that of the brain's highly parallelized and compiled automatic processing (i.e., system one).

Modern humans evolved from earlier people between 200,000 and 50,000 years ago, but numbers and numerical calculation did not exist until about 3400 BCE, when people in Mesopotamia (modern-day Iraq) invented and started using a number system in commerce. By then, the human brain was more or less as it is today. Since the modern human brain evolved before numerical calculation existed, it is not optimized for calculation.

Calculation is done mainly in system two, the brain's controlled, monitored mode. It consumes scarce resources of attention and short-term memory, so when we try to perform calculations entirely in our heads, we have trouble. The exception is that some steps in a calculation may be memorized and therefore are automatic. For example, the overall process of multiplying 479×832 is controlled, but certain substeps of the process may be automatic if we have memorized the multiplication tables for single-digit numbers.

Problems and calculations that involve only one or two steps, in which some steps are memorized (automatic) or don't involve much information, or for which all relevant information is immediately available—and therefore need not be kept in short-term memory—are easy for most people to work out in their heads. For example:

- $9 \times 10 = ?$
- I need to move the washing machine out of the garage, but the car is in the way and my car keys are in my pocket. What to do?
- My girlfriend has two brothers, Bob and Fred. I have met Fred, and the one here now isn't Fred, so it must be Bob.

However, our brains strain to solve problems that require the engagement of system two, especially problems that exceed our short-term memory limits, require certain information to be retrieved from long-term memory, or in which we encounter distractions. For example:

- I need to move the washing machine out of the garage, but the car is in the way, and my car keys are … hmmm … they're not in my pocket. Where are they? … [Search car.] They're not in the car. Maybe I left them in my jacket.… Now where did I leave my jacket? [Search house; eventually find jacket in bedroom; check jacket pockets.] Okay, found the keys … Boy is this bedroom messy—must clean it before wife gets home.… Hmmm. Why did I need the car keys? [Return to garage, see washer.] Oh, yeah; to move the car so I can move the washing machine out of the garage. *(Interim subgoals grabbed attention away from higher-level goals in short-term memory.)*
- Chapter 8 gave examples of tasks in which people have to remember to complete cleanup steps after achieving their primary goal—for example, remembering to remove the last page of a document from a copier after you have the copy.
- John's cat is not black and likes milk. Sue's cat is not brown and doesn't like milk. Sam's cat is not white and doesn't like milk. Mary's cat is not yellow and likes milk. Someone found a cat that is yellow and likes milk. Whose cat is it?[9] *(The negations create more chunks of information than most people's short-term memory can hold at once.)*

[9]Answers provided at the end of this chapter.

- A man built a four-sided house. All four walls faced south. A bear walked by. What color was the bear? *(Requires deduction and knowing and retrieving specific facts about the world and its wildlife.)*
- If five factory workers can assemble five cars in 5 hours, how long does it take 100 factory workers to assemble 100 cars? *(System one offers a quick guess that system two is tempted to accept, but finding the correct answer requires rejecting that guess and engaging system two.)*
- Fred likes classic cars. He doesn't care much about the environment but wants to reduce his gasoline costs. He replaces his '56 Cadillac (12 mpg) with a '57 Chevy (14 mpg). Susan is an environmental activist. She decides to replace her Honda Fit (30 mpg) with a Toyota Prius (40 mpg). If they each drive 10,000 miles over the next year, who saves the most gas? *(Same as previous problem.)*
- You have to measure exactly 4 L of water, but you only have a 3 L bottle and 5 L bottle. How do you do it? *(Requires engaging system two to mentally simulate a series of pours until the right series is found, straining short-term memory and perhaps exceeding mental simulation abilities.)*

When solving such problems, people often use external memory aids, such as writing down interim results, sketching diagrams, and manipulating models of the problem. Such tools augment our limited short-term memory and limited ability to imagine manipulating problem elements. In using such tools, we are distributing the "thinking," off-loading some of it to tools and methods outside of ourselves that are better at certain aspects of cognition than our brains are.

Problem-solving and calculation are also difficult if they require a cognitive strategy, solution method, or procedure that we don't know and cannot devise or find. For example:

- 93.3 × 102.1 = ? *(Requires arithmetic that exceeds short-term memory capacity so must be done with a calculator or on paper. The latter requires knowing how to multiply multidigit decimal numbers on paper.)*
- A farmer has cows and chickens—30 animals total. The animals have a total of 74 legs. How many of each animal does the farmer have? *(Requires translation to two equations and then solving using algebra.)*
- A Zen master blindfolded three of his students. He told them that he would paint either a red or blue dot on each one's forehead. In fact, he painted a red dot on all three foreheads. Then he said, "In a minute I will remove your blindfolds. When I do, look at each other and if you see at least one red dot, raise your hand. Then guess which color your own dot is." Then he removed the blindfolds. The three students looked at each other, then all three raised a hand. After a minute, one of the students said, "My dot is red." How did she know? *(Requires reasoning by contradiction, a specialized method taught in logic and mathematics.)*

- You play a YouTube video on your computer, but there is no sound even though you can see people speaking. Is the problem with the video, video player, computer, speaker cables, or speakers? *(Requires devising and executing a series of diagnostic tests that successively narrow the possible causes of the problem, which requires computer and electronics domain knowledge.)*

These made-up examples demonstrate that certain problems and calculations require training that many people do not have. The sidebar on the next page gives real examples of people being unable to resolve technical problems because they lack training in effective diagnosis in the technical problem domain and are not interested in learning how to do it.

SOLVING TECHNICAL PROBLEMS REQUIRES TECHNICAL INTEREST AND TRAINING

Software engineers are trained to do systematic diagnoses of problems. It is part of their job to know how to devise and execute a series of tests to eliminate possible causes of a fault until they find the cause. Engineers often design technology-based products as if the intended users of the technology were just as skilled as engineers in technical problem diagnosis. However, most people who are not software engineers have not been trained in that sort of problem diagnosis and therefore cannot do it effectively. Here are true examples of nontechnical people facing problems they could not solve without help:

- A friend wanted to book a flight but couldn't because the airline website wouldn't let her. It demanded a password but she didn't have one. She called a software engineer friend, who asked several questions to learn her situation. It turned out that the website assumed she was her husband, because he had previously bought tickets from that airline on that computer. The site wanted his login and password. She didn't know his password and he was out of town. The engineer told her to log out of the website, then return as a *new* customer and create her own account.
- San Francisco's Freecycle Network uses a Yahoo Group. Some people try to join but fail because they cannot figure out how

(Continued)

to complete the registration process. Therefore, they cannot participate in the group.

- At a church, one of two stage-monitor speakers in the audio system stopped working. The assistant music director assumed one of the speakers had failed and said he would replace it. A musician who also is an engineer wasn't sure the speaker was bad, so he swapped the two monitor speaker cables at the speaker end. Now the "bad" speaker worked and the "good" one didn't, showing that the problem was *not* a bad speaker. The assistant music director concluded that one speaker cable was bad and said he would buy a new one. Before he did, the engineer-musician swapped the monitor cables where they connected to the Left and Right outputs of the amplifier, and now the original "bad" speaker again didn't produce sound. The problem was a loose connection in the monitor amplifier output jack.

Even when people know they *could* solve a problem or perform a calculation if they put effort into it, sometimes they don't because they don't consider the potential reward worth the effort. This response is especially common when solving a problem is not required by one's job or otherwise. Some real examples:

- A posting on San Francisco Freecycle Network: "Free: Epson Stylus C86 printer. Was working fine, then suddenly it couldn't recognize the new full ink cartridge. Not sure if it's the cartridge or the printer. So I bought a new printer and am giving the old one away."
- Fred and Alice, a schoolteacher and a nurse who are married, never install or update software on their home computer. They don't know how, and they don't want to know. They use only the software that came with the computer. If their computer says updates are available, they ignore it. If an application (e.g., a web browser) stops working because it is outdated, they stop using it. When necessary, they buy a new computer.
- A 55-year-old woman in the United States has an iPhone. She texts and FaceTimes with her adult son, who lives in England, at least once a day. She took a photo and wanted to send it to her son, but didn't know how. A local friend tried to show her how

to attach the photo to a text or an email, but she considered it too complicated and said she would just wait until the next time her son visited and show him the photo then.

The nontechnical people in these examples are not stupid. Most have college degrees, putting them in the top 30% of educational attainment in the United States. Some are even trained to diagnose problems in different domains (e.g., medicine). They just have no training and/or interest in solving technical problems.

As mentioned above, human cognition is not all in our heads. It is *distributed*, or more accurately, we distribute our cognition. Throughout human history, people have used objects in their environment—sometimes inventing them—to assist in all sorts of cognitive tasks: remembering things, calculating things, and figuring things out. For example:

- counting and recording by making marks with a stick on the ground, with a pen on paper, or with a keyboard on a computer screen;
- inventing arithmetic and mathematics to allow us to calculate accurately;
- inventing alphabets and writing, then the printing press, and later the Internet and Web to allow information to be shared more widely;
- creating social networks to facilitate communicating with other people[10];
- inventing calculators and computers as tools for performing calculations, supporting decision making, and solving problems that humans cannot easily, reliably, or accurately solve on their own;
- designing software applications that support people in performing specific tasks.

Distributing human cognition using external tools can make us more effective thinkers, assuming the tools are well designed. The challenge for designers is to create digital products and services that really support and augment human cognition rather than hindering or burdening it.

IMPORTANT TAKEAWAYS: IMPLICATIONS FOR USER-INTERFACE DESIGN

People often intentionally challenge and entertain themselves by creating or solving puzzles that strain—or "exercise"—their minds (see Fig. 10.3). However, that does not

[10]Some people argue that social networks facilitate communication *too* well.

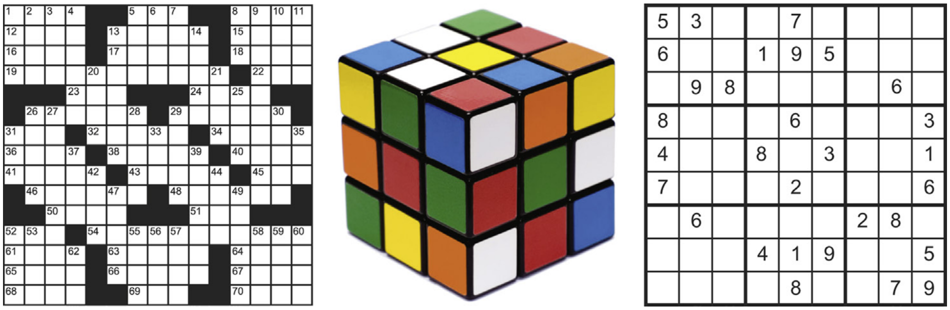


FIGURE 10.3

We challenge ourselves by creating and solving puzzles that tax our mental abilities.

imply that people will happily accept mind-straining problems foisted on them by someone or something else. People have their own goals. They use digital products and services to help them achieve a goal. They want—and need—to focus their attention on that goal. Interactive systems—and designers of them—should respect that and not distract users by imposing technical problems and goals that users don't want.

Here are some examples of technical problems that computers and web services impose on their users:

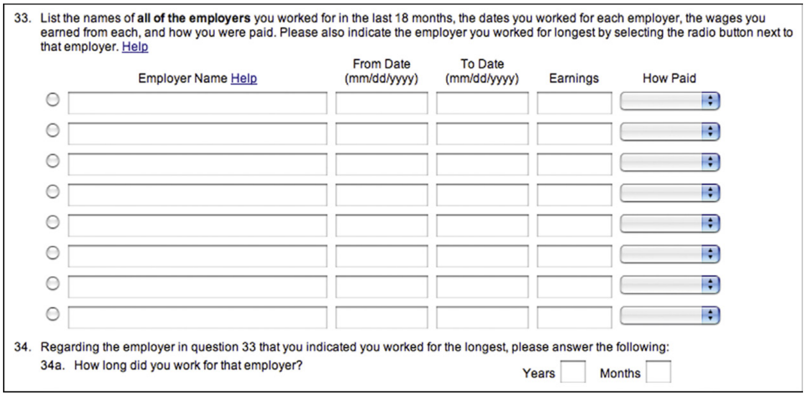
- "It wants my 'member ID.' Is that the same as my 'username'? It must be." *(Requires reasoning by process of elimination.)*
- "Huh? This website charged me the full price! It didn't give me my discount. What now?" *(Requires backtracking through the purchase process to try to figure out what happened.)*
- "I want page numbers in the chapter to start at 23 instead of 1, but I don't see a command to do that. I've tried Page Setup, Document Layout, and View Header and Footer, but it isn't there. All that's left is this Insert Page Numbers command. But I don't want to *insert* page numbers—the chapter already has page numbers. I just want to change the starting number." *(Requires searching methodically through the application's menus and dialogue boxes for a way to change the starting page number, and if it isn't found, deciding by process of elimination that the Insert Page Numbers command must be the way.)*
- "Hmmm. This checkbox is labeled 'Align icons horizontally.' I wonder what happens if I uncheck it. Will my icons be aligned vertically, or will they simply not be aligned?" *(Requires setting the property to ON to see what will happen.)*

Interactive systems should minimize the amount of attention users must devote to operating them (Krug, 2014), because that pulls a person's limited cognitive resources away from the task that user came to the computer to perform. Here are some design rules:

- ***Prominently indicate system status and users' progress toward their goals***. If users can always check their status easily by direct perception, using the system will not strain their attention and short-term memory.
- ***Guide users toward their goals***. Designers can do this implicitly by making sure every choice-point provides a clear information "scent" that leads users toward their goal, or explicitly by using a wizard (multistep dialogue box). Don't just display a bunch of options that appear equally likely and expect users to know how to start toward and get to their goal, especially if they won't perform the task very often.
- ***Tell users explicitly and exactly what they need to know***. Don't expect them to deduce information. Don't require them to figure things out by a process of elimination.
- ***Don't make users diagnose system problems***. For example, a user shouldn't have to diagnose a faulty network connection. Such diagnosis requires technical training that most users don't have.
- ***Minimize the number and complexity of settings***. Don't expect people to optimize combinations of many interacting settings or parameters. People are really bad at that.
- ***Let people use perception rather than calculation***. Some problems that might seem to require calculation can be represented graphically, allowing people to achieve their goals with quick perceptual estimates instead of calculation (see Chapter 12). A simple example: Suppose you want to go to the middle of a document. Document-editing software of the early 1980s forced you to see how many pages the document had, divide that number in half, and issue a command to go to the middle page number. With modern-day document-editing software, you just scroll down until the scrollbar "elevator" is in the middle of the bar, and you are there. Similarly, snap-to grids and alignment guides in drawing tools eliminate the need for users to determine, match, and compute the coordinates of existing graphic elements when adding new ones.
- ***Design for familiarity***. Use concepts, terminology, and graphics that users already know to make the system as familiar to them as possible, requiring them to think about it less. Designers can use this approach to a certain extent even if the system provides functionality that users have not seen before. One way to do this is to follow industry conventions and standards (e.g., Apple Computer, 2020; Microsoft Corporation, 2018). A second way is to make new software applications work like older ones that users have used before. A third approach is to base the design on metaphors, such as the desktop metaphor (Johnson et al., 1989). Finally, designers can study users to learn what is and is not familiar to them.
- ***Design for consistency***. People learn to operate software faster when the design is consistent across the software's various screens, pages, and functions.

This means not only the visual design but also the functionality, task flow, and required user actions (e.g., keystrokes or spoken commands). People also learn software faster if its design is consistent with similar software they already use. See Chapter 12 for more detail on this.

- ***Let the computer do the math***. Don't make people calculate things the computer can calculate itself (see Fig. 10.4).


33. List the names of **all of the employers** you worked for in the last 18 months, the dates you worked for each employer, the wages you earned from each, and how you were paid. Please also indicate the employer you worked for longest by selecting the radio button next to that employer. Help

	Employer Name Help	From Date (mm/dd/yyyy)	To Date (mm/dd/yyyy)	Earnings	How Paid
○					
○					
○					
○					
○					
○					
○					
○					

34. Regarding the employer in question 33 that you indicated you worked for the longest, please answer the following:

34a. How long did you work for that employer? Years Months

FIGURE 10.4

California's online unemployment form asks for data it could calculate itself in both these questions.

ANSWERS TO PUZZLES

- The cat is John's.
- The bear was white, because to have four south-facing walls, the house must be on the North Pole.
- Five workers can assemble five cars in 5 h, so one worker can assemble one car in 5 h and 100 workers can assemble 100 cars in the same 5 h (Adapted from Kahneman, 2011.)
- Fred cuts his gas consumption from 833 gallons to 714 gallons, saving 119 gallons. Susan cuts her gas usage from 333 gallons to 250 gallons, saving 83 gallons. (Adapted from Kahneman, 2011.)
- To end with 4 L of water, fill the 3 L bottle and pour it into the 5 L bottle, then fill the 3 L bottle again and fill up the 5 L bottle. That leaves 1 L in the 3 L bottle. Empty the 5 L bottle and pour the 1 L from the 3 L bottle into the 5 L bottle. Then fill the 3 L bottle again and pour it into the 5 L bottle.

- Let A be the number of cows and B the number of chickens. "A farmer has cows and chicken—30 animals total" translates to $A+B=30$. "The animals have a total of 74 legs" translates to $4A+2B=74$. Solving for A and B gives $A=7$ and $B=23$, so the farmer has 7 cows and 23 chickens.
- The Zen student saw three hands up and red dots on both other students. From this information, she didn't know whether her dot was red or blue. She started out assuming it was blue, and waited. She reasoned that the other students would see her (assumed) blue dot and one other red dot, realize that two red dots were required for all three hands to be up, and quickly figure out that their own dots had to be red. But after a minute neither of the other students had said anything, which told the Zen student that the other students couldn't figure out what color their dots were, which meant that her own dot was *not* blue; it had to be red.