CHAPTER FIVE

HUMAN ERROR? NO, BAD DESIGN

Most industrial accidents are caused by human error: estimates range between 75 and 95 percent. How is it that so many people are so incompetent? Answer: They aren't. It's a design problem.

If the number of accidents blamed upon human error were 1 to 5 percent, I might believe that people were at fault. But when the percentage is so high, then clearly other factors must be involved. When something happens this frequently, there must be another underlying factor.

When a bridge collapses, we analyze the incident to find the causes of the collapse and reformulate the design rules to ensure that form of accident will never happen again. When we discover that electronic equipment is malfunctioning because it is responding to unavoidable electrical noise, we redesign the circuits to be more tolerant of the noise. But when an accident is thought to be caused by people, we blame them and then continue to do things just as we have always done.

Physical limitations are well understood by designers; mental limitations are greatly misunderstood. We should treat all failures in the same way: find the fundamental causes and redesign the system so that these can no longer lead to problems. We design

equipment that requires people to be fully alert and attentive for hours, or to remember archaic, confusing procedures even if they are only used infrequently, sometimes only once in a lifetime. We put people in boring environments with nothing to do for hours on end, until suddenly they must respond quickly and accurately. Or we subject them to complex, high-workload environments, where they are continually interrupted while having to do multiple tasks simultaneously. Then we wonder why there is failure.

Even worse is that when I talk to the designers and administrators of these systems, they admit that they too have nodded off while supposedly working. Some even admit to falling asleep for an instant while driving. They admit to turning the wrong stove burners on or off in their homes, and to other small but significant errors. Yet when their workers do this, they blame them for "human error." And when employees or customers have similar issues, they are blamed for not following the directions properly, or for not being fully alert and attentive.

Understanding Why There Is Error

Error occurs for many reasons. The most common is in the nature of the tasks and procedures that require people to behave in unnatural ways—staying alert for hours at a time, providing precise, accurate control specifications, all the while multitasking, doing several things at once, and subjected to multiple interfering activities. Interruptions are a common reason for error, not helped by designs and procedures that assume full, dedicated attention yet that do not make it easy to resume operations after an interruption. And finally, perhaps the worst culprit of all, is the attitude of people toward errors.

When an error causes a financial loss or, worse, leads to an injury or death, a special committee is convened to investigate the cause and, almost without fail, guilty people are found. The next step is to blame and punish them with a monetary fine, or by firing or jailing them. Sometimes a lesser punishment is proclaimed: make the guilty parties go through more training. Blame and punish; blame and train. The investigations and resulting punishments feel

good: "We caught the culprit." But it doesn't cure the problem: the same error will occur over and over again. Instead, when an error happens, we should determine why, then redesign the product or the procedures being followed so that it will never occur again or, if it does, so that it will have minimal impact.

ROOT CAUSE ANALYSIS

Root cause analysis is the name of the game: investigate the accident until the single, underlying cause is found. What this ought to mean is that when people have indeed made erroneous decisions or actions, we should determine what caused them to err. This is what root cause analysis ought to be about. Alas, all too often it stops once a person is found to have acted inappropriately.

Trying to find the cause of an accident sounds good but it is flawed for two reasons. First, most accidents do not have a single cause: there are usually multiple things that went wrong, multiple events that, had any one of them not occurred, would have prevented the accident. This is what James Reason, the noted British authority on human error, has called the "Swiss cheese model of accidents" (shown in Figure 5.3 of this chapter on page 208, and discussed in more detail there).

Second, why does the root cause analysis stop as soon as a human error is found? If a machine stops working, we don't stop the analysis when we discover a broken part. Instead, we ask: "Why did the part break? Was it an inferior part? Were the required specifications too low? Did something apply too high a load on the part?" We keep asking questions until we are satisfied that we understand the reasons for the failure: then we set out to remedy them. We should do the same thing when we find human error: We should discover what led to the error. When root cause analysis discovers a human error in the chain, its work has just begun: now we apply the analysis to understand why the error occurred, and what can be done to prevent it.

One of the most sophisticated airplanes in the world is the US Air Force's F-22. However, it has been involved in a number of accidents, and pilots have complained that they suffered oxygen

deprivation (hypoxia). In 2010, a crash destroyed an F-22 and killed the pilot. The Air Force investigation board studied the incident and two years later, in 2012, released a report that blamed the accident on pilot error: "failure to recognize and initiate a timely dive recovery due to channelized attention, breakdown of visual scan and unrecognized spatial distortion."

In 2013, the Inspector General's office of the US Department of Defense reviewed the Air Force's findings, disagreeing with the assessment. In my opinion, this time a proper root cause analysis was done. The Inspector General asked "why sudden incapacitation or unconsciousness was not considered a contributory factor." The Air Force, to nobody's surprise, disagreed with the criticism. They argued that they had done a thorough review and that their conclusion "was supported by clear and convincing evidence." Their only fault was that the report "could have been more clearly written."

It is only slightly unfair to parody the two reports this way:

Air Force: It was pilot error—the pilot failed to take corrective action.
Inspector General: That's because the pilot was probably unconscious.
Air Force: So you agree, the pilot failed to correct the problem.

THE FIVE WHYS

Root cause analysis is intended to determine the underlying cause of an incident, not the proximate cause. The Japanese have long followed a procedure for getting at root causes that they call the "Five Whys," originally developed by Sakichi Toyoda and used by the Toyota Motor Company as part of the Toyota Production System for improving quality. Today it is widely deployed. Basically, it means that when searching for the reason, even after you have found one, do not stop: ask why that was the case. And then ask why again. Keep asking until you have uncovered the true underlying causes. Does it take exactly five? No, but calling the procedure "Five Whys" emphasizes the need to keep going even after a reason has been found. Consider how this might be applied to the analysis of the F-22 crash:

Five Whys

Question	Answer
Q1: Why did the plane crash?	Because it was in an uncontrolled dive.
Q2: Why didn't the pilot recover from the dive?	Because the pilot failed to initiate a timely recovery.
Q3: Why was that?	Because he might have been unconscious (or oxygen deprived).
Q4: Why was that?	We don't know. We need to find out.
Etc.	

The Five Whys of this example are only a partial analysis. For example, we need to know why the plane was in a dive (the report explains this, but it is too technical to go into here; suffice it to say that it, too, suggests that the dive was related to a possible oxygen deprivation).

The Five Whys do not guarantee success. The question *why* is ambiguous and can lead to different answers by different investigators. There is still a tendency to stop too soon, perhaps when the limit of the investigator's understanding has been reached. It also tends to emphasize the need to find a single cause for an incident, whereas most complex events have multiple, complex causal factors. Nonetheless, it is a powerful technique.

The tendency to stop seeking reasons as soon as a human error has been found is widespread. I once reviewed a number of accidents in which highly trained workers at an electric utility company had been electrocuted when they contacted or came too close to the high-voltage lines they were servicing. All the investigating committees found the workers to be at fault, something even the workers (those who had survived) did not dispute. But when the committees were investigating the complex causes of the incidents, why did they stop once they found a human error? Why didn't they keep going to find out why the error had occurred, what circumstances had led to it, and then, why those circumstances had happened? The committees never went far enough to find the deeper, root causes of the accidents. Nor did they consider redesigning the systems and procedures to make the incidents

either impossible or far less likely. When people err, change the system so that type of error will be reduced or eliminated. When complete elimination is not possible, redesign to reduce the impact.

It wasn't difficult for me to suggest simple changes to procedures that would have prevented most of the incidents at the utility company. It had never occurred to the committee to think of this. The problem is that to have followed my recommendations would have meant changing the culture from an attitude among the field workers that "We are supermen: we can solve any problem, repair the most complex outage. We do not make errors." It is not possible to eliminate human error if it is thought of as a personal failure rather than as a sign of poor design of procedures or equipment. My report to the company executives was received politely. I was even thanked. Several years later I contacted a friend at the company and asked what changes they had made. "No changes," he said. "And we are still injuring people."

One big problem is that the natural tendency to blame someone for an error is even shared by those who made the error, who often agree that it was their fault. People do tend to blame themselves when they do something that, after the fact, seems inexcusable. "I knew better," is a common comment by those who have erred. But when someone says, "It was my fault, I knew better," this is not a valid analysis of the problem. That doesn't help prevent its recurrence. When many people all have the same problem, shouldn't another cause be found? If the system lets you make the error, it is badly designed. And if the system induces you to make the error, then it is really badly designed. When I turn on the wrong stove burner, it is not due to my lack of knowledge: it is due to poor mapping between controls and burners. Teaching me the relationship will not stop the error from recurring: redesigning the stove will.

We can't fix problems unless people admit they exist. When we blame people, it is then difficult to convince organizations to restructure the design to eliminate these problems. After all, if a person is at fault, replace the person. But seldom is this the case: usually the system, the procedures, and social pressures have led

to the problems, and the problems won't be fixed without addressing all of these factors.

Why do people err? Because the designs focus upon the requirements of the system and the machines, and not upon the requirements of people. Most machines require precise commands and guidance, forcing people to enter numerical information perfectly. But people aren't very good at great precision. We frequently make errors when asked to type or write sequences of numbers or letters. This is well known: so why are machines still being designed that require such great precision, where pressing the wrong key can lead to horrendous results?

People are creative, constructive, exploratory beings. We are particularly good at novelty, at creating new ways of doing things, and at seeing new opportunities. Dull, repetitive, precise requirements fight against these traits. We are alert to changes in the environment, noticing new things, and then thinking about them and their implications. These are virtues, but they get turned into negative features when we are forced to serve machines. Then we are punished for lapses in attention, for deviating from the tightly prescribed routines.

A major cause of error is time stress. Time is often critical, especially in such places as manufacturing or chemical processing plants and hospitals. But even everyday tasks can have time pressures. Add environmental factors, such as poor weather or heavy traffic, and the time stresses increase. In commercial establishments, there is strong pressure not to slow the processes, because doing so would inconvenience many, lead to significant loss of money, and, in a hospital, possibly decrease the quality of patient care. There is a lot of pressure to push ahead with the work even when an outside observer would say it was dangerous to do so. In many industries, if the operators actually obeyed all the procedures, the work would never get done. So we push the boundaries: we stay up far longer than is natural. We try to do too many tasks at the same time. We drive faster than is safe. Most of the time we manage okay. We might even be rewarded and praised for our he-

roic efforts. But when things go wrong and we fail, then this same behavior is blamed and punished.

Deliberate Violations

Errors are not the only type of human failures. Sometimes people knowingly take risks. When the outcome is positive, they are often rewarded. When the result is negative, they might be punished. But how do we classify these deliberate violations of known, proper behavior? In the error literature, they tend to be ignored. In the accident literature, they are an important component.

Deliberate deviations play an important role in many accidents. They are defined as cases where people intentionally violate procedures and regulations. Why do they happen? Well, almost every one of us has probably deliberately violated laws, rules, or even our own best judgment at times. Ever go faster than the speed limit? Drive too fast in the snow or rain? Agree to do some hazardous act, even while privately thinking it foolhardy to do so?

In many industries, the rules are written more with a goal toward legal compliance than with an understanding of the work requirements. As a result, if workers followed the rules, they couldn't get their jobs done. Do you sometimes prop open locked doors? Drive with too little sleep? Work with co-workers even though you are ill (and might therefore be infectious)?

Routine violations occur when noncompliance is so frequent that it is ignored. Situational violations occur when there are special circumstances (example: going through a red light "because no other cars were visible and I was late"). In some cases, the only way to complete a job might be to violate a rule or procedure.

A major cause of violations is inappropriate rules or procedures that not only invite violation but encourage it. Without the violations, the work could not be done. Worse, when employees feel it necessary to violate the rules in order to get the job done and, as a result, succeed, they will probably be congratulated and rewarded. This, of course, unwittingly rewards noncompliance. Cultures that encourage and commend violations set poor role models.

Although violations are a form of error, these are organizational and societal errors, important but outside the scope of the design of everyday things. The human error examined here is unintentional: deliberate violations, by definition, are intentional deviations that are known to be risky, with the potential of doing harm.

Two Types of Errors: Slips and Mistakes

Many years ago, the British psychologist James Reason and I developed a general classification of human error. We divided human error into two major categories: slips and mistakes (Figure 5.1). This classification has proved to be of value for both theory and practice. It is widely used in the study of error in such diverse areas as industrial and aviation accidents, and medical errors. The discussion gets a little technical, so I have kept technicalities to a minimum. This topic is of extreme importance to design, so stick with it.

DEFINITIONS: ERRORS, SLIPS, AND MISTAKES

Human error is defined as any deviance from "appropriate" behavior. The word *appropriate* is in quotes because in many circumstances, the appropriate behavior is not known or is only deter-

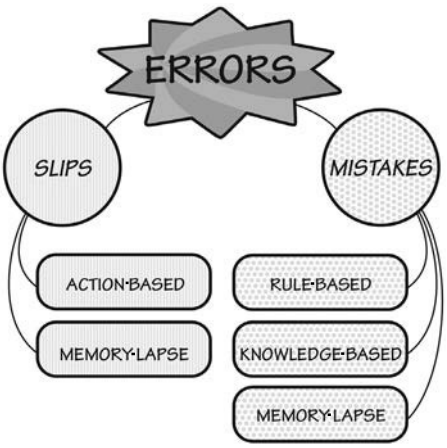

FIGURE 5.1. Classification of Errors. Errors have two major forms. Slips occur when the goal is correct, but the required actions are not done properly: the execution is flawed. Mistakes occur when the goal or plan is wrong. Slips and mistakes can be further divided based upon their underlying causes. Memory lapses can lead to either slips or mistakes, depending upon whether the memory failure was at the highest level of cognition (mistakes) or at lower (subconscious) levels (slips). Although deliberate violations of procedures are clearly inappropriate behaviors that often lead to accidents, these are not considered as errors (see discussion in text).

mined after the fact. But still, error is defined as deviance from the generally accepted correct or appropriate behavior.

Error is the general term for all wrong actions. There are two major classes of error: *slips* and *mistakes*, as shown in Figure 5.1; slips are further divided into two major classes and mistakes into three. These categories of errors all have different implications for design. I now turn to a more detailed look at these classes of errors and their design implications.

SLIPS

A slip occurs when a person intends to do one action and ends up doing something else. With a slip, the action performed is not the same as the action that was intended.

There are two major classes of slips: *action-based* and *memory-lapse.* In action-based slips, the wrong action is performed. In lapses, memory fails, so the intended action is not done or its results not evaluated. Action-based slips and memory lapses can be further classified according to their causes.

Example of an action-based slip. I poured some milk into my coffee and then put the coffee cup into the refrigerator. This is the correct action applied to the wrong object.

Example of a memory-lapse slip. I forget to turn off the gas burner on my stove after cooking dinner.

MISTAKES

A mistake occurs when the wrong goal is established or the wrong plan is formed. From that point on, even if the actions are executed properly they are part of the error, because the actions themselves are inappropriate—they are part of the wrong plan. With a mistake, the action that is performed matches the plan: it is the plan that is wrong.

Mistakes have three major classes: *rule-based, knowledge-based,* and *memory-lapse.* In a rule-based mistake, the person has appropriately diagnosed the situation, but then decided upon an erroneous course of action: the wrong rule is being followed. In a knowledge-based mistake, the problem is misdiagnosed because

of erroneous or incomplete knowledge. Memory-lapse mistakes take place when there is forgetting at the stages of goals, plans, or evaluation. Two of the mistakes leading to the "Gimli Glider" Boeing 767 emergency landing were:

Example of knowledge-based mistake. Weight of fuel was computed in pounds instead of kilograms.

Example of memory-lapse mistake. A mechanic failed to complete troubleshooting because of distraction.

ERROR AND THE SEVEN STAGES OF ACTION

Errors can be understood through reference to the seven stages of the action cycle of Chapter 2 (Figure 5.2). Mistakes are errors in setting the goal or plan, and in comparing results with expectations—the higher levels of cognition. Slips happen in the execution of a plan, or in the perception or interpretation of the outcome—the lower stages. Memory lapses can happen at any of the eight transitions between stages, shown by the X's in Figure 5.2B. A memory lapse at one of these transitions stops the action cycle from proceeding, and so the desired action is not completed.

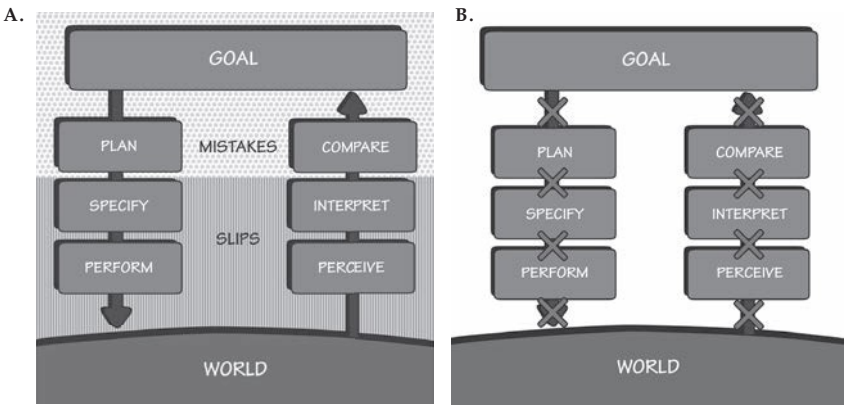

FIGURE 5.2. Where Slips and Mistakes Originate in the Action Cycle. Figure A shows that action slips come from the bottom four stages of the action cycle and mistakes from the top three stages. Memory lapses impact the transitions between stages (shown by the X's in Figure B). Memory lapses at the higher levels lead to mistakes, and lapses at the lower levels lead to slips.

Slips are the result of subconscious actions getting waylaid en route. Mistakes result from conscious deliberations. The same processes that make us creative and insightful by allowing us to see relationships between apparently unrelated things, that let us leap to correct conclusions on the basis of partial or even faulty evidence, also lead to mistakes. Our ability to generalize from small amounts of information helps tremendously in new situations; but sometimes we generalize too rapidly, classifying a new situation as similar to an old one when, in fact, there are significant discrepancies. This leads to mistakes that can be difficult to discover, let alone eliminate.

The Classification of Slips

> *A colleague reported that he went to his car to drive to work. As he drove away, he realized that he had forgotten his briefcase, so he turned around and went back. He stopped the car, turned off the engine, and unbuckled his wristwatch. Yes, his wristwatch, instead of his seatbelt.*

The story illustrates both a memory-lapse slip and an action slip. The forgetting of the briefcase is a memory-lapse slip. The unbuckling of the wristwatch is an action slip, in this case a combination of description-similarity and capture error (described later in this chapter).

Most everyday errors are slips. Intending to do one action, you find yourself doing another. When a person says something clearly and distinctly to you, you "hear" something quite different. The study of slips is the study of the psychology of everyday errors—what Freud called "the psychopathology of everyday life." Freud believed that slips have hidden, dark meanings, but most are accounted for by rather simple mental mechanisms.

An interesting property of slips is that, paradoxically, they tend to occur more frequently to skilled people than to novices. Why? Because slips often result from a lack of attention to the task. Skilled people—experts—tend to perform tasks automatically, under subconscious control. Novices have to pay considerable conscious attention, resulting in a relatively low occurrence of slips.

Some slips result from the similarities of actions. Or an event in the world may automatically trigger an action. Sometimes our thoughts and actions may remind us of unintended actions, which we then perform. There are numerous different kinds of action slips, categorized by the underlying mechanisms that give rise to them. The three most relevant to design are:

- capture slips
- description-similarity slips
- mode errors

CAPTURE SLIPS

> *I was using a copying machine, and I was counting the pages. I found myself counting, "1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King." I had been playing cards recently.*

The capture slip is defined as the situation where, instead of the desired activity, a more frequently or recently performed one gets done instead: it captures the activity. Capture errors require that part of the action sequences involved in the two activities be identical, with one sequence being far more familiar than the other. After doing the identical part, the more frequent or more recent activity continues, and the intended one does not get done. Seldom, if ever, does the unfamiliar sequence capture the familiar one. All that is needed is a lapse of attention to the desired action at the critical junction when the identical portions of the sequences diverge into the two different activities. Capture errors are, therefore, partial memory-lapse errors. Interestingly, capture errors are more prevalent in experienced skilled people than in beginners, in part because the experienced person has automated the required actions and may not be paying conscious attention when the intended action deviates from the more frequent one.

Designers need to avoid procedures that have identical opening steps but then diverge. The more experienced the workers, the more likely they are to fall prey to capture. Whenever possible, sequences should be designed to differ from the very start.

DESCRIPTION-SIMILARITY SLIPS

A former student reported that one day he came home from jogging, took off his sweaty shirt, and rolled it up in a ball, intending to throw it in the laundry basket. Instead he threw it in the toilet. (It wasn't poor aim: the laundry basket and toilet were in different rooms.)

In the slip known as a description-similarity slip, the error is to act upon an item similar to the target. This happens when the description of the target is sufficiently vague. Much as we saw in Chapter 3, Figure 3.1, where people had difficulty distinguishing among different images of money because their internal descriptions did not have sufficient discriminating information, the same thing can happen to us, especially when we are tired, stressed, or overloaded. In the example that opened this section, both the laundry basket and the toilet bowl are containers, and if the description of the target was sufficiently ambiguous, such as "a large enough container," the slip could be triggered.

Remember the discussion in Chapter 3 that most objects don't need precise descriptions, simply enough precision to distinguish the desired target from alternatives. This means that a description that usually suffices may fail when the situation changes so that multiple similar items now match the description. Description-similarity errors result in performing the correct action on the wrong object. Obviously, the more the wrong and right objects have in common, the more likely the errors are to occur. Similarly, the more objects present at the same time, the more likely the error.

Designers need to ensure that controls and displays for different purposes are significantly different from one another. A lineup of identical-looking switches or displays is very apt to lead to description-similarity error. In the design of airplane cockpits, many controls are shape coded so that they both look and feel different from one another: the throttle levers are different from the flap levers (which might look and feel like a wing flap), which are different from the landing gear control (which might look and feel like a wheel).

MEMORY-LAPSE SLIPS

Errors caused by memory failures are common. Consider these examples:

- Making copies of a document, walking off with the copy, but leaving the original inside the machine.
- Forgetting a child. This error has numerous examples, such as leaving a child behind at a rest stop during a car trip, or in the dressing room of a department store, or a new mother forgetting her one-month-old and having to go to the police for help in finding the baby.
- Losing a pen because it was taken out to write something, then put down while doing some other task. The pen is forgotten in the activities of putting away a checkbook, picking up goods, talking to a salesperson or friends, and so on. Or the reverse: borrowing a pen, using it, and then putting it away in your pocket or purse, even though it is someone else's (this is also a capture error).
- Using a bank or credit card to withdraw money from an automatic teller machine, then walking off without the card, is such a frequent error that many machines now have a forcing function: the card must be removed before the money will be delivered. Of course, it is then possible to walk off without the money, but this is less likely than forgetting the card because money is the goal of using the machine.

Memory lapses are common causes of error. They can lead to several kinds of errors: failing to do all of the steps of a procedure; repeating steps; forgetting the outcome of an action; or forgetting the goal or plan, thereby causing the action to be stopped.

The immediate cause of most memory-lapse failures is interruptions, events that intervene between the time an action is decided upon and the time it is completed. Quite often the interference comes from the machines we are using: the many steps required between the start and finish of the operations can overload the capacity of short-term or working memory.

There are several ways to combat memory-lapse errors. One is to minimize the number of steps; another, to provide vivid reminders of steps that need to be completed. A superior method is to use the

forcing function of Chapter 4. For example, automated teller machines often require removal of the bank card before delivering the requested money: this prevents forgetting the bank card, capitalizing on the fact that people seldom forget the goal of the activity, in this case the money. With pens, the solution is simply to prevent their removal, perhaps by chaining public pens to the counter. Not all memory-lapse errors lend themselves to simple solutions. In many cases the interruptions come from outside the system, where the designer has no control.

MODE-ERROR SLIPS

A mode error occurs when a device has different states in which the same controls have different meanings: we call these states *modes.* Mode errors are inevitable in anything that has more possible actions than it has controls or displays; that is, the controls mean different things in the different modes. This is unavoidable as we add more and more functions to our devices.

Ever turn off the wrong device in your home entertainment system? This happens when one control is used for multiple purposes. In the home, this is simply frustrating. In industry, the confusion that results when operators believe the system to be in one mode, when in reality it is in another, has resulted in serious accidents and loss of life.

It is tempting to save money and space by having a single control serve multiple purposes. Suppose there are ten different functions on a device. Instead of using ten separate knobs or switches—which would take considerable space, add extra cost, and appear intimidatingly complex, why not use just two controls, one to select the function, the other to set the function to the desired condition? Although the resulting design appears quite simple and easy to use, this apparent simplicity masks the underlying complexity of use. The operator must always be completely aware of the mode, of what function is active. Alas, the prevalence of mode errors shows this assumption to be false. Yes, if I select a mode and then immediately adjust the parameters, I am not apt to be confused about the state. But what if I select the mode and then get interrupted

by other events? Or if the mode is maintained for considerable periods? Or, as in the case of the Airbus accident discussed below, the two modes being selected are very similar in control and function, but have different operating characteristics, which means that the resulting mode error is difficult to discover? Sometimes the use of modes is justifiable, such as the need to put many controls and displays in a small, restricted space, but whatever the reason, modes are a common cause of confusion and error.

Alarm clocks often use the same controls and display for setting the time of day and the time the alarm should go off, and many of us have thereby set one when we meant the other. Similarly, when time is displayed on a twelve-hour scale, it is easy to set the alarm to go off at seven A.M. only later to discover that the alarm had been set for seven P.M. The use of "A.M." and "P.M." to distinguish times before and after noon is a common source of confusion and error, hence the common use of 24-hour time specification throughout most of the world (the major exceptions being North America, Australia, India, and the Philippines). Watches with multiple functions have similar problems, in this case required because of the small amount of space available for controls and displays. Modes exist in most computer programs, in our cell phones, and in the automatic controls of commercial aircraft. A number of serious accidents in commercial aviation can be attributed to mode errors, especially in aircraft that use automatic systems (which have a large number of complex modes). As automobiles become more complex, with the dashboard controls for driving, heating and air-conditioning, entertainment, and navigation, modes are increasingly common.

An accident with an Airbus airplane illustrates the problem. The flight control equipment (often referred to as the automatic pilot) had two modes, one for controlling vertical speed, the other for controlling the flight path's angle of descent. In one case, when the pilots were attempting to land, the pilots thought that they were controlling the angle of descent, whereas they had accidentally

selected the mode that controlled speed of descent. The number (–3.3) that was entered into the system to represent an appropriate angle (–3.3°) was too steep a rate of descent when interpreted as vertical speed (–3,300 feet/minute: –3.3° would only be –800 feet/minute). This mode confusion contributed to the resulting fatal accident. After a detailed study of the accident, Airbus changed the display on the instrument so that vertical speed would always be displayed with a four-digit number and angle with two digits, thus reducing the chance of confusion.

Mode error is really design error. Mode errors are especially likely where the equipment does not make the mode visible, so the user is expected to remember what mode has been established, sometimes hours earlier, during which time many intervening events might have occurred. Designers must try to avoid modes, but if they are necessary, the equipment must make it obvious which mode is invoked. Once again, designers must always compensate for interfering activities.

The Classification of Mistakes

Mistakes result from the choice of inappropriate goals and plans or from faulty comparison of the outcome with the goals during evaluation. In mistakes, a person makes a poor decision, misclassifies a situation, or fails to take all the relevant factors into account. Many mistakes arise from the vagaries of human thought, often because people tend to rely upon remembered experiences rather than on more systematic analysis. We make decisions based upon what is in our memory. But as discussed in Chapter 3, retrieval from long-term memory is actually a reconstruction rather than an accurate record. As a result, it is subject to numerous biases. Among other things, our memories tend to be biased toward overgeneralization of the commonplace and overemphasis of the discrepant.

The Danish engineer Jens Rasmussen distinguished among three modes of behavior: skill-based, rule-based, and knowledge-based. This three-level classification scheme provides a practical tool that has found wide acceptance in applied areas, such as the design of

many industrial systems. Skill-based behavior occurs when workers are extremely expert at their jobs, so they can do the everyday, routine tasks with little or no thought or conscious attention. The most common form of errors in skill-based behavior is slips.

Rule-based behavior occurs when the normal routine is no longer applicable but the new situation is one that is known, so there is already a well-prescribed course of action: a rule. Rules simply might be learned behaviors from previous experiences, but includes formal procedures prescribed in courses and manuals, usually in the form of "if-then" statements, such as, "*If* the engine will not start, *then* do [the appropriate action]." Errors with rule-based behavior can be either a mistake or a slip. If the wrong rule is selected, this would be a mistake. If the error occurs during the execution of the rule, it is most likely a slip.

Knowledge-based procedures occur when unfamiliar events occur, where neither existing skills nor rules apply. In this case, there must be considerable reasoning and problem-solving. Plans might be developed, tested, and then used or modified. Here, conceptual models are essential in guiding development of the plan and interpretation of the situation.

In both rule-based and knowledge-based situations, the most serious mistakes occur when the situation is misdiagnosed. As a result, an inappropriate rule is executed, or in the case of knowledge-based problems, the effort is addressed to solving the wrong problem. In addition, with misdiagnosis of the problem comes misinterpretation of the environment, as well as faulty comparisons of the current state with expectations. These kinds of mistakes can be very difficult to detect and correct.

RULE-BASED MISTAKES

When new procedures have to be invoked or when simple problems arise, we can characterize the actions of skilled people as rule-based. Some rules come from experience; others are formal procedures in manuals or rulebooks, or even less formal guides, such as cookbooks for food preparation. In either case, all we must do is identify the situation, select the proper rule, and then follow it.

When driving, behavior follows well-learned rules. Is the light red? If so, stop the car. Wish to turn left? Signal the intention to turn and move as far left as legally permitted: slow the vehicle and wait for a safe break in traffic, all the while following the traffic rules and relevant signs and lights.

Rule-based mistakes occur in multiple ways:

- The situation is mistakenly interpreted, thereby invoking the wrong goal or plan, leading to following an inappropriate rule.
- The correct rule is invoked, but the rule itself is faulty, either because it was formulated improperly or because conditions are different than assumed by the rule or through incomplete knowledge used to determine the rule. All of these lead to knowledge-based mistakes.
- The correct rule is invoked, but the outcome is incorrectly evaluated. This error in evaluation, usually rule- or knowledge-based itself, can lead to further problems as the action cycle continues.

Example 1: In 2013, at the Kiss nightclub in Santa Maria, Brazil, pyrotechnics used by the band ignited a fire that killed over 230 people. The tragedy illustrates several mistakes. The band made a knowledge-based mistake when they used outdoor flares, which ignited the ceiling's acoustic tiles. The band thought the flares were safe. Many people rushed into the rest rooms, mistakenly thinking they were exits: they died. Early reports suggested that the guards, unaware of the fire, at first mistakenly blocked people from leaving the building. Why? Because nightclub attendees would sometimes leave without paying for their drinks.

The mistake was in devising a rule that did not take account of emergencies. A root cause analysis would reveal that the goal was to prevent inappropriate exit but still allow the doors to be used in an emergency. One solution is doors that trigger alarms when used, deterring people trying to sneak out, but allowing exit when needed.

Example 2: Turning the thermostat of an oven to its maximum temperature to get it to the proper cooking temperature faster is a mistake based upon a false conceptual model of the way the oven works. If the person wanders off and forgets to come back and check the oven

temperature after a reasonable period (a memory-lapse slip), the improper high setting of the oven temperature can lead to an accident, possibly a fire.

Example 3: A driver, unaccustomed to anti-lock brakes, encounters an unexpected object in the road on a wet, rainy day. The driver applies full force to the brakes but the car skids, triggering the anti-lock brakes to rapidly turn the brakes on and off, as they are designed to do. The driver, feeling the vibrations, believes that it indicates malfunction and therefore lifts his foot off the brake pedal. In fact, the vibration is a signal that anti-lock brakes are working properly. The driver's misevaluation leads to the wrong behavior.

Rule-based mistakes are difficult to avoid and then difficult to detect. Once the situation has been classified, the selection of the appropriate rule is often straightforward. But what if the classification of the situation is wrong? This is difficult to discover because there is usually considerable evidence to support the erroneous classification of the situation and the choice of rule. In complex situations, the problem is too much information: information that both supports the decision and also contradicts it. In the face of time pressures to make a decision, it is difficult to know which evidence to consider, which to reject. People usually decide by taking the current situation and matching it with something that happened earlier. Although human memory is quite good at matching examples from the past with the present situation, this doesn't mean that the matching is accurate or appropriate. The matching is biased by recency, regularity, and uniqueness. Recent events are remembered far better than less recent ones. Frequent events are remembered through their regularities, and unique events are remembered because of their uniqueness. But suppose the current event is different from all that has been experienced before: people are still apt to find some match in memory to use as a guide. The same powers that make us so good at dealing with the common and the unique lead to severe error with novel events.

What is a designer to do? Provide as much guidance as possible to ensure that the current state of things is displayed in a coherent

and easily interpreted format—ideally graphical. This is a difficult problem. All major decision makers worry about the complexity of real-world events, where the problem is often too much information, much of it contradictory. Often, decisions must be made quickly. Sometimes it isn't even clear that there is an incident or that a decision is actually being made.

Think of it like this. In your home, there are probably a number of broken or misbehaving items. There might be some burnt-out lights, or (in my home) a reading light that works fine for a little while, then goes out: we have to walk over and wiggle the fluorescent bulb. There might be a leaky faucet or other minor faults that you know about but are postponing action to remedy. Now consider a major process-control manufacturing plant (an oil refinery, a chemical plant, or a nuclear power plant). These have thousands, perhaps tens of thousands, of valves and gauges, displays and controls, and so on. Even the best of plants always has some faulty parts. The maintenance crews always have a list of items to take care of. With all the alarms that trigger when a problem arises, even though it might be minor, and all the everyday failures, how does one know which might be a significant indicator of a major problem? Every single one usually has a simple, rational explanation, so not making it an urgent item is a sensible decision. In fact, the maintenance crew simply adds it to a list. Most of the time, this is the correct decision. The one time in a thousand (or even, one time in a million) that the decision is wrong makes it the one they will be blamed for: how could they have missed such obvious signals?

Hindsight is always superior to foresight. When the accident investigation committee reviews the event that contributed to the problem, they know what actually happened, so it is easy for them to pick out which information was relevant, which was not. This is retrospective decision making. But when the incident was taking place, the people were probably overwhelmed with far too much irrelevant information and probably not a lot of relevant information. How were they to know which to attend to and which to ignore? Most of the time, experienced operators get things right. The one time they fail, the retrospective analysis is apt to condemn

them for missing the obvious. Well, during the event, nothing may be obvious. I return to this topic later in the chapter.

You will face this while driving, while handling your finances, and while just going through your daily life. Most of the unusual incidents you read about are not relevant to you, so you can safely ignore them. Which things should be paid attention to, which should be ignored? Industry faces this problem all the time, as do governments. The intelligence communities are swamped with data. How do they decide which cases are serious? The public hears about their mistakes, but not about the far more frequent cases that they got right or about the times they ignored data as not being meaningful—and were correct to do so.

If every decision had to be questioned, nothing would ever get done. But if decisions are not questioned, there will be major mistakes—rarely, but often of substantial penalty.

The design challenge is to present the information about the state of the system (a device, vehicle, plant, or activities being monitored) in a way that is easy to assimilate and interpret, as well as to provide alternative explanations and interpretations. It is useful to question decisions, but impossible to do so if every action—or failure to act—requires close attention.

This is a difficult problem with no obvious solution.

KNOWLEDGE-BASED MISTAKES

Knowledge-based behavior takes place when the situation is novel enough that there are no skills or rules to cover it. In this case, a new procedure must be devised. Whereas skills and rules are controlled at the behavioral level of human processing and are therefore subconscious and automatic, knowledge-based behavior is controlled at the reflective level and is slow and conscious.

With knowledge-based behavior, people are consciously problem solving. They are in an unknown situation and do not have any available skills or rules that apply directly. Knowledge-based behavior is required either when a person encounters an unknown situation, perhaps being asked to use some novel equipment, or

even when doing a familiar task and things go wrong, leading to a novel, uninterpretable state.

The best solution to knowledge-based situations is to be found in a good understanding of the situation, which in most cases also translates into an appropriate conceptual model. In complex cases, help is needed, and here is where good cooperative problem-solving skills and tools are required. Sometimes, good procedural manuals (paper or electronic) will do the job, especially if critical observations can be used to arrive at the relevant procedures to follow. A more powerful approach is to develop intelligent computer systems, using good search and appropriate reasoning techniques (artificial-intelligence decision-making and problem-solving). The difficulties here are in establishing the interaction of the people with the automation: human teams and automated systems have to be thought of as collaborative, cooperative systems. Instead, they are often built by assigning the tasks that machines can do to the machines and leaving the humans to do the rest. This usually means that machines do the parts that are easy for people, but when the problems become complex, which is precisely when people could use assistance, that is when the machines usually fail. (I discuss this problem extensively in *The Design of Future Things*.)

MEMORY-LAPSE MISTAKES

Memory lapses can lead to mistakes if the memory failure leads to forgetting the goal or plan of action. A common cause of the lapse is an interruption that leads to forgetting the evaluation of the current state of the environment. These lead to mistakes, not slips, because the goals and plans become wrong. Forgetting earlier evaluations often means remaking the decision, sometimes erroneously.

The design cures for memory-lapse mistakes are the same as for memory-lapse slips: ensure that all the relevant information is continuously available. The goals, plans, and current evaluation of the system are of particular importance and should be continually available. Far too many designs eliminate all signs of these items once they have been made or acted upon. Once again, the designer

should assume that people will be interrupted during their activities and that they may need assistance in resuming their operations.

Social and Institutional Pressures

A subtle issue that seems to figure in many accidents is social pressure. Although at first it may not seem relevant to design, it has strong influence on everyday behavior. In industrial settings, social pressures can lead to misinterpretation, mistakes, and accidents. To understand human error, it is essential to understand social pressure.

Complex problem-solving is required when one is faced with knowledge-based problems. In some cases, it can take teams of people days to understand what is wrong and the best ways to respond. This is especially true of situations where mistakes have been made in the diagnosis of the problem. Once the mistaken diagnosis is made, all information from then on is interpreted from the wrong point of view. Appropriate reconsiderations might only take place during team turnover, when new people come into the situation with a fresh viewpoint, allowing them to form different interpretations of the events. Sometimes just asking one or more of the team members to take a few hours' break can lead to the same fresh analysis (although it is understandably difficult to convince someone who is battling an emergency situation to stop for a few hours).

In commercial installations, the pressure to keep systems running is immense. Considerable money might be lost if an expensive system is shut down. Operators are often under pressure not to do this. The result has at times been tragic. Nuclear power plants are kept running longer than is safe. Airplanes have taken off before everything was ready and before the pilots had received permission. One such incident led to the largest accident in aviation history. Although the incident happened in 1977, a long time ago, the lessons learned are still very relevant today.

In Tenerife, in the Canary Islands, a KLM Boeing 747 crashed during takeoff into a Pan American 747 that was taxiing on the same runway, killing 583 people. The KLM plane had not received clearance to take off, but the weather was starting to get bad and the crew had already been delayed for too long (even being on the

Canary Islands was a diversion from the scheduled flight—bad weather had prevented their landing at their scheduled destination). And the Pan American flight should not have been on the runway, but there was considerable misunderstanding between the pilots and the air traffic controllers. Furthermore, the fog was coming in so thickly that neither plane's crew could see the other.

In the Tenerife disaster, time and economic pressures were acting together with cultural and weather conditions. The Pan American pilots questioned their orders to taxi on the runway, but they continued anyway. The first officer of the KLM flight voiced minor objections to the captain, trying to explain that they were not yet cleared for takeoff (but the first officer was very junior to the captain, who was one of KLM's most respected pilots). All in all, a major tragedy occurred due to a complex mixture of social pressures and logical explaining away of discrepant observations.

You may have experienced similar pressure, putting off refueling or recharging your car until it was too late and you ran out, sometimes in a truly inconvenient place (this has happened to me). What are the social pressures to cheat on school examinations, or to help others cheat? Or to not report cheating by others? Never underestimate the power of social pressures on behavior, causing otherwise sensible people to do things they know are wrong and possibly dangerous.

When I was in training to do underwater (scuba) diving, our instructor was so concerned about this that he said he would reward anyone who stopped a dive early in favor of safety. People are normally buoyant, so they need weights to get them beneath the surface. When the water is cold, the problem is intensified because divers must then wear either wet or dry suits to keep warm, and these suits add buoyancy. Adjusting buoyancy is an important part of the dive, so along with the weights, divers also wear air vests into which they continually add or remove air so that the body is close to neutral buoyancy. (As divers go deeper, increased water pressure compresses the air in their protective suits and lungs, so they become heavier: the divers need to add air to their vests to compensate.)

When divers have gotten into difficulties and needed to get to the surface quickly, or when they were at the surface close to shore but being tossed around by waves, some drowned because they were still being encumbered by their heavy weights. Because the weights are expensive, the divers didn't want to release them. In addition, if the divers released the weights and then made it back safely, they could never prove that the release of the weights was necessary, so they would feel embarrassed, creating self-induced social pressure. Our instructor was very aware of the resulting reluctance of people to take the critical step of releasing their weights when they weren't entirely positive it was necessary. To counteract this tendency, he announced that if anyone dropped the weights for safety reasons, he would publicly praise the diver and replace the weights at no cost to the person. This was a very persuasive attempt to overcome social pressures.

Social pressures show up continually. They are usually difficult to document because most people and organizations are reluctant to admit these factors, so even if they are discovered in the process of the accident investigation, the results are often kept hidden from public scrutiny. A major exception is in the study of transportation accidents, where the review boards across the world tend to hold open investigations. The US National Transportation Safety Board (NTSB) is an excellent example of this, and its reports are widely used by many accident investigators and researchers of human error (including me).

Another good example of social pressures comes from yet another airplane incident. In 1982 an Air Florida flight from National Airport, Washington, DC, crashed during takeoff into the Fourteenth Street Bridge over the Potomac River, killing seventy-eight people, including four who were on the bridge. The plane should not have taken off because there was ice on the wings, but it had already been delayed for over an hour and a half; this and other factors, the NTSB reported, "may have predisposed the crew to hurry." The accident occurred despite the first officer's attempt to warn the captain, who was flying the airplane (the captain and first officer—sometimes called the copilot—usually alternate flying

roles on different legs of a trip). The NTSB report quotes the flight deck recorder's documenting that "although the first officer expressed concern that something 'was not right' to the captain four times during the takeoff, the captain took no action to reject the takeoff." NTSB summarized the causes this way:

> *The National Transportation Safety Board determines that the probable cause of this accident was the flight crew's failure to use engine anti-ice during ground operation and takeoff, their decision to take off with snow/ice on the airfoil surfaces of the aircraft, and the captain's failure to reject the takeoff during the early stage when his attention was called to anomalous engine instrument readings.* (NTSB, 1982.)

Again we see social pressures coupled with time and economic forces.

Social pressures can be overcome, but they are powerful and pervasive. We drive when drowsy or after drinking, knowing full well the dangers, but talking ourselves into believing that we are exempt. How can we overcome these kinds of social problems? Good design alone is not sufficient. We need different training; we need to reward safety and put it above economic pressures. It helps if the equipment can make the potential dangers visible and explicit, but this is not always possible. To adequately address social, economic, and cultural pressures and to improve upon company policies are the hardest parts of ensuring safe operation and behavior.

CHECKLISTS

Checklists are powerful tools, proven to increase the accuracy of behavior and to reduce error, particularly slips and memory lapses. They are especially important in situations with multiple, complex requirements, and even more so where there are interruptions. With multiple people involved in a task, it is essential that the lines of responsibility be clearly spelled out. It is always better to have two people do checklists together as a team: one to read the instruction, the other to execute it. If, instead, a single person executes the checklist and then, later, a second person checks the items, the

results are not as robust. The person following the checklist, feeling confident that any errors would be caught, might do the steps too quickly. But the same bias affects the checker. Confident in the ability of the first person, the checker often does a quick, less than thorough job.

One paradox of groups is that quite often, adding more people to check a task makes it less likely that it will be done right. Why? Well, if you were responsible for checking the correct readings on a row of fifty gauges and displays, but you know that two people before you had checked them and that one or two people who come after you will check your work, you might relax, thinking that you don't have to be extra careful. After all, with so many people looking, it would be impossible for a problem to exist without detection. But if everyone thinks the same way, adding more checks can actually increase the chance of error. A collaboratively followed checklist is an effective way to counteract these natural human tendencies.

In commercial aviation, collaboratively followed checklists are widely accepted as essential tools for safety. The checklist is done by two people, usually the two pilots of the airplane (the captain and first officer). In aviation, checklists have proven their worth and are now required in all US commercial flights. But despite the strong evidence confirming their usefulness, many industries still fiercely resist them. It makes people feel that their competence is being questioned. Moreover, when two people are involved, a junior person (in aviation, the first officer) is being asked to watch over the action of the senior person. This is a strong violation of the lines of authority in many cultures.

Physicians and other medical professionals have strongly resisted the use of checklists. It is seen as an insult to their professional competence. "Other people might need checklists," they complain, "but not me." Too bad. Too err is human: we all are subject to slips and mistakes when under stress, or under time or social pressure, or after being subjected to multiple interruptions, each essential in its own right. It is not a threat to professional competence to be

human. Legitimate criticisms of particular checklists are used as an indictment against the concept of checklists. Fortunately, checklists are slowly starting to gain acceptance in medical situations. When senior personnel insist on the use of checklists, it actually enhances their authority and professional status. It took decades for checklists to be accepted in commercial aviation: let us hope that medicine and other professions will change more rapidly.

Designing an effective checklist is difficult. The design needs to be iterative, always being refined, ideally using the human-centered design principles of Chapter 6, continually adjusting the list until it covers the essential items yet is not burdensome to perform. Many people who object to checklists are actually objecting to badly designed lists: designing a checklist for a complex task is best done by professional designers in conjunction with subject matter experts.

Printed checklists have one major flaw: they force the steps to follow a sequential ordering, even where this is not necessary or even possible. With complex tasks, the order in which many operations are performed may not matter, as long as they are all completed. Sometimes items early in the list cannot be done at the time they are encountered in the checklist. For example, in aviation one of the steps is to check the amount of fuel in the plane. But what if the fueling operation has not yet been completed when this checklist item is encountered? Pilots will skip over it, intending to come back to it after the plane has been refueled. This is a clear opportunity for a memory-lapse error.

In general, it is bad design to impose a sequential structure to task execution unless the task itself requires it. This is one of the major benefits of electronic checklists: they can keep track of skipped items and can ensure that the list will not be marked as complete until all items have been done.

Reporting Error

If errors can be caught, then many of the problems they might lead to can often be avoided. But not all errors are easy to detect. Moreover, social pressures often make it difficult for people to admit to

their own errors (or to report the errors of others). If people report their own errors, they might be fined or punished. Moreover, their friends may make fun of them. If a person reports that someone else made an error, this may lead to severe personal repercussions. Finally, most institutions do not wish to reveal errors made by their staff. Hospitals, courts, police systems, utility companies—all are reluctant to admit to the public that their workers are capable of error. These are all unfortunate attitudes.

The only way to reduce the incidence of errors is to admit their existence, to gather together information about them, and thereby to be able to make the appropriate changes to reduce their occurrence. In the absence of data, it is difficult or impossible to make improvements. Rather than stigmatize those who admit to error, we should thank those who do so and encourage the reporting. We need to make it easier to report errors, for the goal is not to punish, but to determine how it occurred and change things so that it will not happen again.

CASE STUDY: *JIDOKA*—HOW TOYOTA HANDLES ERROR

The Toyota automobile company has developed an extremely efficient error-reduction process for manufacturing, widely known as the Toyota Production System. Among its many key principles is a philosophy called *Jidoka*, which Toyota says is "roughly translated as 'automation with a human touch.'" If a worker notices something wrong, the worker is supposed to report it, sometimes even stopping the entire assembly line if a faulty part is about to proceed to the next station. (A special cord, called an *andon*, stops the assembly line and alerts the expert crew.) Experts converge upon the problem area to determine the cause. "Why did it happen?" "Why was that?" "Why is that the reason?" The philosophy is to ask "Why?" as many times as may be necessary to get to the root cause of the problem and then fix it so it can never occur again.

As you might imagine, this can be rather discomforting for the person who found the error. But the report is expected, and when it is discovered that people have failed to report errors, they are punished, all in an attempt to get the workers to be honest.

POKA-YOKE: ERROR PROOFING

Poka-yoke is another Japanese method, this one invented by Shigeo Shingo, one of the Japanese engineers who played a major role in the development of the Toyota Production System. *Poka-yoke* translates as "error proofing" or "avoiding error." One of the techniques of poka-yoke is to add simple fixtures, jigs, or devices to constrain the operations so that they are correct. I practice this myself in my home. One trivial example is a device to help me remember which way to turn the key on the many doors in the apartment complex where I live. I went around with a pile of small, circular, green stick-on dots and put them on each door beside its keyhole, with the green dot indicating the direction in which the key needed to be turned: I added signifiers to the doors. Is this a major error? No. But eliminating it has proven to be convenient. (Neighbors have commented on their utility, wondering who put them there.)

In manufacturing facilities, poka-yoke might be a piece of wood to help align a part properly, or perhaps plates designed with asymmetrical screw holes so that the plate could fit in only one position. Covering emergency or critical switches with a cover to prevent accidental triggering is another poka-yoke technique: this is obviously a forcing function. All the poka-yoke techniques involve a combination of the principles discussed in this book: affordances, signifiers, mapping, and constraints, and perhaps most important of all, forcing functions.

NASA'S AVIATION SAFETY REPORTING SYSTEM

US commercial aviation has long had an extremely effective system for encouraging pilots to submit reports of errors. The program has resulted in numerous improvements to aviation safety. It wasn't easy to establish: pilots had severe self-induced social pressures against admitting to errors. Moreover, to whom would they report them? Certainly not to their employers. Not even to the Federal Aviation Authority (FAA), for then they would probably be punished. The solution was to let the National Aeronautics and Space Administration (NASA) set up a voluntary accident reporting system whereby pilots could submit semi-anonymous reports

of errors they had made or observed in others (semi-anonymous because pilots put their name and contact information on the reports so that NASA could call to request more information). Once NASA personnel had acquired the necessary information, they would detach the contact information from the report and mail it back to the pilot. This meant that NASA no longer knew who had reported the error, which made it impossible for the airline companies or the FAA (which enforced penalties against errors) to find out who had submitted the report. If the FAA had independently noticed the error and tried to invoke a civil penalty or certificate suspension, the receipt of self-report automatically exempted the pilot from punishment (for minor infractions).

When a sufficient number of similar errors had been collected, NASA would analyze them and issue reports and recommendations to the airlines and to the FAA. These reports also helped the pilots realize that their error reports were valuable tools for increasing safety. As with checklists, we need similar systems in the field of medicine, but it has not been easy to set up. NASA is a neutral body, charged with enhancing aviation safety, but has no oversight authority, which helped gain the trust of pilots. There is no comparable institution in medicine: physicians are afraid that self-reported errors might lead them to lose their license or be subjected to lawsuits. But we can't eliminate errors unless we know what they are. The medical field is starting to make progress, but it is a difficult technical, political, legal, and social problem.

Detecting Error

Errors do not necessarily lead to harm if they are discovered quickly. The different categories of errors have differing ease of discovery. In general, action slips are relatively easy to discover; mistakes, much more difficult. Action slips are relatively easy to detect because it is usually easy to notice a discrepancy between the intended act and the one that got performed. But this detection can only take place if there is feedback. If the result of the action is not visible, how can the error be detected?

Memory-lapse slips are difficult to detect precisely because there is nothing to see. With a memory slip, the required action is not performed. When no action is done, there is nothing to detect. It is only when the lack of action allows some unwanted event to occur that there is hope of detecting a memory-lapse slip.

Mistakes are difficult to detect because there is seldom anything that can signal an inappropriate goal. And once the wrong goal or plan is decided upon, the resulting actions are consistent with that wrong goal, so careful monitoring of the actions not only fails to detect the erroneous goal, but, because the actions are done correctly, can inappropriately provide added confidence to the decision.

Faulty diagnoses of a situation can be surprisingly difficult to detect. You might expect that if the diagnosis was wrong, the actions would turn out to be ineffective, so the fault would be discovered quickly. But misdiagnoses are not random. Usually they are based on considerable knowledge and logic. The misdiagnosis is usually both reasonable and relevant to eliminating the symptoms being observed. As a result, the initial actions are apt to appear appropriate and helpful. This makes the problem of discovery even more difficult. The actual error might not be discovered for hours or days.

Memory-lapse mistakes are especially difficult to detect. Just as with a memory-lapse slip the absence of something that should have been done is always more difficult to detect than the presence of something that should not have been done. The difference between memory-lapse slips and mistakes is that, in the first case, a single component of a plan is skipped, whereas in the second, the entire plan is forgotten. Which is easier to discover? At this point I must retreat to the standard answer science likes to give to questions of this sort: "It all depends."

EXPLAINING AWAY MISTAKES

Mistakes can take a long time to be discovered. Hear a noise that sounds like a pistol shot and think: "Must be a car's exhaust backfiring." Hear someone yell outside and think: "Why can't my

neighbors be quiet?" Are we correct in dismissing these incidents? Most of the time we are, but when we're not, our explanations can be difficult to justify.

Explaining away errors is a common problem in commercial accidents. Most major accidents are preceded by warning signs: equipment malfunctions or unusual events. Often, there is a series of apparently unrelated breakdowns and errors that culminate in major disaster. Why didn't anyone notice? Because no single incident appeared to be serious. Often, the people involved noted each problem but discounted it, finding a logical explanation for the otherwise deviant observation.

THE CASE OF THE WRONG TURN ON A HIGHWAY

I've misinterpreted highway signs, as I'm sure most drivers have. My family was traveling from San Diego to Mammoth Lakes, California, a ski area about 400 miles north. As we drove, we noticed more and more signs advertising the hotels and gambling casinos of Las Vegas, Nevada. "Strange," we said, "Las Vegas always did advertise a long way off—there is even a billboard in San Diego—but this seems excessive, advertising on the road to Mammoth." We stopped for gasoline and continued on our journey. Only later, when we tried to find a place to eat supper, did we discover that we had missed a turn nearly two hours earlier, before we had stopped for gasoline, and that we were actually on the road to Las Vegas, not the road to Mammoth. We had to backtrack the entire two-hour segment, wasting four hours of driving. It's humorous now; it wasn't then.

Once people find an explanation for an apparent anomaly, they tend to believe they can now discount it. But explanations are based on analogy with past experiences, experiences that may not apply to the current situation. In the driving story, the prevalence of billboards for Las Vegas was a signal we should have heeded, but it seemed easily explained. Our experience is typical: some major industrial incidents have resulted from false explanations of anomalous events. But do note: usually these apparent anomalies should be ignored. Most of the time, the explanation for their pres-

ence is correct. Distinguishing a true anomaly from an apparent one is difficult.

IN HINDSIGHT, EVENTS SEEM LOGICAL

The contrast in our understanding before and after an event can be dramatic. The psychologist Baruch Fischhoff has studied explanations given in hindsight, where events seem completely obvious and predictable after the fact but completely unpredictable beforehand.

Fischhoff presented people with a number of situations and asked them to predict what would happen: they were correct only at the chance level. When the actual outcome was not known by the people being studied, few predicted the actual outcome. He then presented the same situations along with the actual outcomes to another group of people, asking them to state how likely each outcome was: when the actual outcome was known, it appeared to be plausible and likely and other outcomes appeared unlikely.

Hindsight makes events seem obvious and predictable. Foresight is difficult. During an incident, there are never clear clues. Many things are happening at once: workload is high, emotions and stress levels are high. Many things that are happening will turn out to be irrelevant. Things that appear irrelevant will turn out to be critical. The accident investigators, working with hindsight, knowing what really happened, will focus on the relevant information and ignore the irrelevant. But at the time the events were happening, the operators did not have information that allowed them to distinguish one from the other.

This is why the best accident analyses can take a long time to do. The investigators have to imagine themselves in the shoes of the people who were involved and consider all the information, all the training, and what the history of similar past events would have taught the operators. So, the next time a major accident occurs, ignore the initial reports from journalists, politicians, and executives who don't have any substantive information but feel compelled to provide statements anyway. Wait until the official reports come from trusted sources. Unfortunately, this could be months or years after the accident, and the public usually wants

answers immediately, even if those answers are wrong. Moreover, when the full story finally appears, newspapers will no longer consider it news, so they won't report it. You will have to search for the official report. In the United States, the National Transportation Safety Board (NTSB) can be trusted. NTSB conducts careful investigations of all major aviation, automobile and truck, train, ship, and pipeline incidents. (Pipelines? Sure: pipelines transport coal, gas, and oil.)

Designing for Error

It is relatively easy to design for the situation where everything goes well, where people use the device in the way that was intended, and no unforeseen events occur. The tricky part is to design for when things go wrong.

Consider a conversation between two people. Are errors made? Yes, but they are not treated as such. If a person says something that is not understandable, we ask for clarification. If a person says something that we believe to be false, we question and debate. We don't issue a warning signal. We don't beep. We don't give error messages. We ask for more information and engage in mutual dialogue to reach an understanding. In normal conversations between two friends, misstatements are taken as normal, as approximations to what was really meant. Grammatical errors, self-corrections, and restarted phrases are ignored. In fact, they are usually not even detected because we concentrate upon the intended meaning, not the surface features.

Machines are not intelligent enough to determine the meaning of our actions, but even so, they are far less intelligent than they could be. With our products, if we do something inappropriate, if the action fits the proper format for a command, the product does it, even if it is outrageously dangerous. This has led to tragic accidents, especially in health care, where inappropriate design of infusion pumps and X-ray machines allowed extreme overdoses of medication or radiation to be administered to patients, leading to their deaths. In financial institutions, simple keyboard errors have led to huge financial transactions, far beyond normal limits.

Even simple checks for reasonableness would have stopped all of these errors. (This is discussed at the end of the chapter under the heading "Sensibility Checks.")

Many systems compound the problem by making it easy to err but difficult or impossible to discover error or to recover from it. It should not be possible for one simple error to cause widespread damage. Here is what should be done:

- Understand the causes of error and design to minimize those causes.
- Do sensibility checks. Does the action pass the "common sense" test?
- Make it possible to reverse actions—to "undo" them—or make it harder to do what cannot be reversed.
- Make it easier for people to discover the errors that do occur, and make them easier to correct.
- Don't treat the action as an error; rather, try to help the person complete the action properly. Think of the action as an approximation to what is desired.

As this chapter demonstrates, we know a lot about errors. Thus, novices are more likely to make mistakes than slips, whereas experts are more likely to make slips. Mistakes often arise from ambiguous or unclear information about the current state of a system, the lack of a good conceptual model, and inappropriate procedures. Recall that most mistakes result from erroneous choice of goal or plan or erroneous evaluation and interpretation. All of these come about through poor information provided by the system about the choice of goals and the means to accomplish them (plans), and poor-quality feedback about what has actually happened.

A major source of error, especially memory-lapse errors, is interruption. When an activity is interrupted by some other event, the cost of the interruption is far greater than the loss of the time required to deal with the interruption: it is also the cost of resuming the interrupted activity. To resume, it is necessary to remember precisely the previous state of the activity: what the goal was, where one was in the action cycle, and the relevant state of the system. Most systems make it difficult to resume after an interruption.

Most discard critical information that is needed by the user to remember the numerous small decisions that had been made, the things that were in the person's short-term memory, to say nothing of the current state of the system. What still needs to be done? Maybe I was finished? It is no wonder that many slips and mistakes are the result of interruptions.

Multitasking, whereby we deliberately do several tasks simultaneously, erroneously appears to be an efficient way of getting a lot done. It is much beloved by teenagers and busy workers, but in fact, all the evidence points to severe degradation of performance, increased errors, and a general lack of both quality and efficiency. Doing two tasks at once takes longer than the sum of the times it would take to do each alone. Even as simple and common a task as talking on a hands-free cell phone while driving leads to serious degradation of driving skills. One study even showed that cell phone usage during walking led to serious deficits: "Cell phone users walked more slowly, changed directions more frequently, and were less likely to acknowledge other people than individuals in the other conditions. In the second study, we found that cell phone users were less likely to notice an unusual activity along their walking route (a unicycling clown)" (Hyman, Boss, Wise, McKenzie, & Caggiano, 2010).

A large percentage of medical errors are due to interruptions. In aviation, where interruptions were also determined to be a major problem during the critical phases of flying—landing and takeoff—the US Federal Aviation Authority (FAA) requires what it calls a "Sterile Cockpit Configuration," whereby pilots are not allowed to discuss any topic not directly related to the control of the airplane during these critical periods. In addition, the flight attendants are not permitted to talk to the pilots during these phases (which has at times led to the opposite error—failure to inform the pilots of emergency situations).

Establishing similar sterile periods would be of great benefit to many professions, including medicine and other safety-critical operations. My wife and I follow this convention in driving: when the driver is entering or leaving a high-speed highway, conversa-

tion ceases until the transition has been completed. Interruptions and distractions lead to errors, both mistakes and slips.

Warning signals are usually not the answer. Consider the control room of a nuclear power plant, the cockpit of a commercial aircraft, or the operating room of a hospital. Each has a large number of different instruments, gauges, and controls, all with signals that tend to sound similar because they all use simple tone generators to beep their warnings. There is no coordination among the instruments, which means that in major emergencies, they all sound at once. Most can be ignored anyway because they tell the operator about something that is already known. Each competes with the others to be heard, interfering with efforts to address the problem.

Unnecessary, annoying alarms occur in numerous situations. How do people cope? By disconnecting warning signals, taping over warning lights (or removing the bulbs), silencing bells, and basically getting rid of all the safety warnings. The problem comes after such alarms are disabled, either when people forget to restore the warning systems (there are those memory-lapse slips again), or if a different incident happens while the alarms are disconnected. At that point, nobody notices. Warnings and safety methods must be used with care and intelligence, taking into account the tradeoffs for the people who are affected.

The design of warning signals is surprisingly complex. They have to be loud or bright enough to be noticed, but not so loud or bright that they become annoying distractions. The signal has to both attract attention (act as a signifier of critical information) and also deliver information about the nature of the event that is being signified. The various instruments need to have a coordinated response, which means that there must be international standards and collaboration among the many design teams from different, often competing, companies. Although considerable research has been directed toward this problem, including the development of national standards for alarm management systems, the problem still remains in many situations.

More and more of our machines present information through speech. But like all approaches, this has both strengths and

weaknesses. It allows for precise information to be conveyed, especially when the person's visual attention is directed elsewhere. But if several speech warnings operate at the same time, or if the environment is noisy, speech warnings may not be understood. Or if conversations among the users or operators are necessary, speech warnings will interfere. Speech warning signals can be effective, but only if used intelligently.

DESIGN LESSONS FROM THE STUDY OF ERRORS

Several design lessons can be drawn from the study of errors, one for preventing errors before they occur and one for detecting and correcting them when they do occur. In general, the solutions follow directly from the preceding analyses.

ADDING CONSTRAINTS TO BLOCK ERRORS

Prevention often involves adding specific constraints to actions. In the physical world, this can be done through clever use of shape and size. For example, in automobiles, a variety of fluids are required for safe operation and maintenance: engine oil, transmission oil, brake fluid, windshield washer solution, radiator coolant, battery water, and gasoline. Putting the wrong fluid into a reservoir could lead to serious damage or even an accident. Automobile manufacturers try to minimize these errors by segregating the filling points, thereby reducing description-similarity errors. When the filling points for fluids that should be added only occasionally or by qualified mechanics are located separately from those for fluids used more frequently, the average motorist is unlikely to use the incorrect filling points. Errors in adding fluids to the wrong container can be minimized by making the openings have different sizes and shapes, providing physical constraints against inappropriate filling. Different fluids often have different colors so that they can be distinguished. All these are excellent ways to minimize errors. Similar techniques are in widespread use in hospitals and industry. All of these are intelligent applications of constraints, forcing functions, and poka-yoke.

Electronic systems have a wide range of methods that could be used to reduce error. One is to segregate controls, so that easily confused controls are located far from one another. Another is to use separate modules, so that any control not directly relevant to the current operation is not visible on the screen, but requires extra effort to get to.

UNDO

Perhaps the most powerful tool to minimize the impact of errors is the Undo command in modern electronic systems, reversing the operations performed by the previous command, wherever possible. The best systems have multiple levels of undoing, so it is possible to undo an entire sequence of actions.

Obviously, undoing is not always possible. Sometimes, it is only effective if done immediately after the action. Still, it is a powerful tool to minimize the impact of error. It is still amazing to me that many electronic and computer-based systems fail to provide a means to undo even where it is clearly possible and desirable.

CONFIRMATION AND ERROR MESSAGES

Many systems try to prevent errors by requiring confirmation before a command will be executed, especially when the action will destroy something of importance. But these requests are usually ill-timed because after requesting an operation, people are usually certain they want it done. Hence the standard joke about such warnings:

Person: Delete "my most important file."
System: Do you want to delete "my most important file"?
Person: Yes.
System: Are you certain?
Person: Yes!
System "My most favorite file" has been deleted.
Person: Oh. Damn.

The request for confirmation seems like an irritant rather than an essential safety check because the person tends to focus upon the action rather than the object that is being acted upon. A better check would be a prominent display of both the action to be taken and the object, perhaps with the choice of "cancel" or "do it." The important point is making salient what the implications of the action are. Of course, it is because of errors of this sort that the Undo command is so important. With traditional graphical user interfaces on computers, not only is Undo a standard command, but when files are "deleted," they are actually simply moved from sight and stored in the file folder named "Trash," so that in the above example, the person could open the Trash and retrieve the erroneously deleted file.

Confirmations have different implications for slips and mistakes. When I am writing, I use two very large displays and a powerful computer. I might have seven to ten applications running simultaneously. I have sometimes had as many as forty open windows. Suppose I activate the command that closes one of the windows, which triggers a confirmatory message: did I wish to close the window? How I deal with this depends upon why I requested that the window be closed. If it was a slip, the confirmation required will be useful. If it was by mistake, I am apt to ignore it. Consider these two examples:

A slip leads me to close the wrong window.

Suppose I intended to type the word *We,* but instead of typing Shift + W for the first character, I typed Command + W (or Control + W), the keyboard command for closing a window. Because I expected the screen to display an uppercase W, when a dialog box appeared, asking whether I really wanted to delete the file, I would be surprised, which would immediately alert me to the slip. I would cancel the action (an alternative thoughtfully provided by the dialog box) and retype the Shift + W, carefully this time.

A mistake leads me to close the wrong window.

Now suppose I really intended to close a window. I often use a temporary file in a window to keep notes about the chapter I am working on. When I am finished with it, I close it without saving its contents—after all, I am finished. But because I usually have multiple windows open, it is very easy to close the wrong one. The computer assumes that all commands apply to the active window—the one where the last actions had been performed (and which contains the text cursor). But if I reviewed the temporary window prior to closing it, my visual attention is focused upon that window, and when I decide to close it, I forget that it is not the active window from the computer's point of view. So I issue the command to shut the window, the computer presents me with a dialog box, asking for confirmation, and I accept it, choosing the option not to save my work. Because the dialog box was expected, I didn't bother to read it. As a result, I closed the wrong window and worse, did not save any of the typing, possibly losing considerable work. Warning messages are surprisingly ineffective against mistakes (even nice requests, such as the one shown in Chapter 4, Figure 4.6, page 143).

Was this a mistake or a slip? Both. Issuing the "close" command while the wrong window was active is a memory-lapse slip. But deciding not to read the dialog box and accepting it without saving the contents is a mistake (two mistakes, actually).

What can a designer do? Several things:

- **Make the item being acted upon more prominent.** That is, change the appearance of the actual object being acted upon to be more visible: enlarge it, or perhaps change its color.
- **Make the operation reversible.** If the person saves the content, no harm is done except the annoyance of having to reopen the file. If the person elects Don't Save, the system could secretly save the contents, and the next time the person opened the file, it could ask whether it should restore it to the latest condition.

SENSIBILITY CHECKS

Electronic systems have another advantage over mechanical ones: they can check to make sure that the requested operation is sensible.

It is amazing that in today's world, medical personnel can accidentally request a radiation dose a thousand times larger than normal and have the equipment meekly comply. In some cases, it isn't even possible for the operator to notice the error.

Similarly, errors in stating monetary sums can lead to disastrous results, even though a quick glance at the amount would indicate that something was badly off. For example, there are roughly 1,000 Korean won to the US dollar. Suppose I wanted to transfer $1,000 into a Korean bank account in *won* ($1,000 is roughly ₩1,000,000). But suppose I enter the Korean number into the dollar field. Oops—I'm trying to transfer a million dollars. Intelligent systems would take note of the normal size of my transactions, querying if the amount was considerably larger than normal. For me, it would query the million-dollar request. Less intelligent systems would blindly follow instructions, even though I did not have a million dollars in my account (in fact, I would probably be charged a fee for overdrawing my account).

Sensibility checks, of course, are also the answer to the serious errors caused when inappropriate values are entered into hospital medication and X-ray systems or in financial transactions, as discussed earlier in this chapter.

MINIMIZING SLIPS

Slips most frequently occur when the conscious mind is distracted, either by some other event or simply because the action being performed is so well learned that it can be done automatically, without conscious attention. As a result, the person does not pay sufficient attention to the action or its consequences. It might therefore seem that one way to minimize slips is to ensure that people always pay close, conscious attention to the acts being done.

Bad idea. Skilled behavior is subconscious, which means it is fast, effortless, and usually accurate. Because it is so automatic, we can type at high speeds even while the conscious mind is occupied composing the words. This is why we can walk and talk while navigating traffic and obstacles. If we had to pay conscious attention to every little thing we did, we would accomplish far less in our

lives. The information processing structures of the brain automatically regulate how much conscious attention is being paid to a task: conversations automatically pause when crossing the street amid busy traffic. Don't count on it, though: if too much attention is focused on something else, the fact that the traffic is getting dangerous might not be noted.

Many slips can be minimized by ensuring that the actions and their controls are as dissimilar as possible, or at least, as physically far apart as possible. Mode errors can be eliminated by the simple expedient of eliminating most modes and, if this is not possible, by making the modes very visible and distinct from one another.

The best way of mitigating slips is to provide perceptible feedback about the nature of the action being performed, then very perceptible feedback describing the new resulting state, coupled with a mechanism that allows the error to be undone. For example, the use of machine-readable codes has led to a dramatic reduction in the delivery of wrong medications to patients. Prescriptions sent to the pharmacy are given electronic codes, so the pharmacist can scan both the prescription and the resulting medication to ensure they are the same. Then, the nursing staff at the hospital scans both the label of the medication and the tag worn around the patient's wrist to ensure that the medication is being given to the correct individual. Moreover, the computer system can flag repeated administration of the same medication. These scans do increase the workload, but only slightly. Other kinds of errors are still possible, but these simple steps have already been proven worthwhile.

Common engineering and design practices seem as if they are deliberately intended to cause slips. Rows of identical controls or meters is a sure recipe for description-similarity errors. Internal modes that are not very conspicuously marked are a clear driver of mode errors. Situations with numerous interruptions, yet where the design assumes undivided attention, are a clear enabler of memory lapses—and almost no equipment today is designed to support the numerous interruptions that so many situations entail. And failure to provide assistance and visible reminders for performing infrequent procedures that are similar to much more

frequent ones leads to capture errors, where the more frequent actions are performed rather than the correct ones for the situation. Procedures should be designed so that the initial steps are as dissimilar as possible.

The important message is that good design can prevent slips and mistakes. Design can save lives.

THE SWISS CHEESE MODEL OF HOW ERRORS LEAD TO ACCIDENTS

Fortunately, most errors do not lead to accidents. Accidents often have numerous contributing causes, no single one of which is the root cause of the incident.

James Reason likes to explain this by invoking the metaphor of multiple slices of Swiss cheese, the cheese famous for being riddled with holes (Figure 5.3). If each slice of cheese represents a condition in the task being done, an accident can happen only if holes in all four slices of cheese are lined up just right. In well-designed systems, there can be many equipment failures, many errors, but they will not lead to an accident unless they all line up precisely. Any leakage—passageway through a hole—is most likely blocked at the next level. Well-designed systems are resilient against failure.

This is why the attempt to find "the" cause of an accident is usually doomed to fail. Accident investigators, the press, government officials, and the everyday citizen like to find simple explanations for the cause of an accident. "See, if the hole in slice A

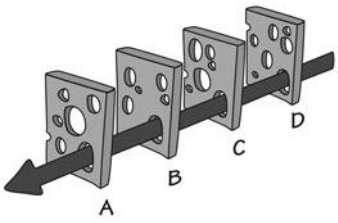

FIGURE 5.3. Reason's Swiss Cheese Model of Accidents. Accidents usually have multiple causes, whereby had any single one of those causes not happened, the accident would not have occurred. The British accident researcher James Reason describes this through the metaphor of slices of Swiss cheese: unless the holes all line up perfectly, there will be no accident. This metaphor provides two lessons: First, do not try to find "the" cause of an accident; Second, we can decrease accidents and make systems more resilient by designing them to have extra precautions against error (more slices of cheese), less opportunities for slips, mistakes, or equipment failure (less holes), and very different mechanisms in the different subparts of the system (trying to ensure that the holes do not line up). (Drawing based upon one by Reason, 1990.)

had been slightly higher, we would not have had the accident. So throw away slice A and replace it." Of course, the same can be said for slices B, C, and D (and in real accidents, the number of cheese slices would sometimes measure in the tens or hundreds). It is relatively easy to find some action or decision that, had it been different, would have prevented the accident. But that does not mean that this was the cause of the accident. It is only one of the many causes: all the items have to line up.

You can see this in most accidents by the "if only" statements. "If only I hadn't decided to take a shortcut, I wouldn't have had the accident." "If only it hadn't been raining, my brakes would have worked." "If only I had looked to the left, I would have seen the car sooner." Yes, all those statements are true, but none of them is "the" cause of the accident. Usually, there is no single cause. Yes, journalists and lawyers, as well as the public, like to know the cause so someone can be blamed and punished. But reputable investigating agencies know that there is not a single cause, which is why their investigations take so long. Their responsibility is to understand the system and make changes that would reduce the chance of the same sequence of events leading to a future accident.

The Swiss cheese metaphor suggests several ways to reduce accidents:

- Add more slices of cheese.
- Reduce the number of holes (or make the existing holes smaller).
- Alert the human operators when several holes have lined up.

Each of these has operational implications. More slices of cheese means mores lines of defense, such as the requirement in aviation and other industries for checklists, where one person reads the items, another does the operation, and the first person checks the operation to confirm it was done appropriately.

Reducing the number of critical safety points where error can occur is like reducing the number or size of the holes in the Swiss cheese. Properly designed equipment will reduce the opportunity for slips and mistakes, which is like reducing the number of holes

and making the ones that remain smaller. This is precisely how the safety level of commercial aviation has been dramatically improved. Deborah Hersman, chair of the National Transportation Safety Board, described the design philosophy as:

> *U.S. airlines carry about two million people through the skies safely every day, which has been achieved in large part through design redundancy and layers of defense.*

Design redundancy and layers of defense: that's Swiss cheese. The metaphor illustrates the futility of trying to find the one underlying cause of an accident (usually some person) and punishing the culprit. Instead, we need to think about systems, about all the interacting factors that lead to human error and then to accidents, and devise ways to make the systems, as a whole, more reliable.

When Good Design Isn't Enough

WHEN PEOPLE REALLY ARE AT FAULT

I am sometimes asked whether it is really right to say that people are never at fault, that it is always bad design. That's a sensible question. And yes, of course, sometimes it is the person who is at fault.

Even competent people can lose competency if sleep deprived, fatigued, or under the influence of drugs. This is why we have laws banning pilots from flying if they have been drinking within some specified period and why we limit the number of hours they can fly without rest. Most professions that involve the risk of death or injury have similar regulations about drinking, sleep, and drugs. But everyday jobs do not have these restrictions. Hospitals often require their staff to go without sleep for durations that far exceed the safety requirements of airlines. Why? Would you be happy having a sleep-deprived physician operating on you? Why is sleep deprivation considered dangerous in one situation and ignored in another?

Some activities have height, age, or strength requirements. Others require considerable skills or technical knowledge: people

not trained or not competent should not be doing them. That is why many activities require government-approved training and licensing. Some examples are automobile driving, airplane piloting, and medical practice. All require instructional courses and tests. In aviation, it isn't sufficient to be trained: pilots must also keep in practice by flying some minimum number of hours per month.

Drunk driving is still a major cause of automobile accidents: this is clearly the fault of the drinker. Lack of sleep is another major culprit in vehicle accidents. But because people occasionally are at fault does not justify the attitude that assumes they are always at fault. The far greater percentage of accidents is the result of poor design, either of equipment or, as is often the case in industrial accidents, of the procedures to be followed.

As noted in the discussion of deliberate violations earlier in this chapter (page 169), people will sometimes deliberately violate procedures and rules, perhaps because they cannot get their jobs done otherwise, perhaps because they believe there are extenuating circumstances, and sometimes because they are taking the gamble that the relatively low probability of failure does not apply to them. Unfortunately, if someone does a dangerous activity that only results in injury or death one time in a million, that can lead to hundreds of deaths annually across the world, with its 7 billion people. One of my favorite examples in aviation is of a pilot who, after experiencing low oil-pressure readings in all three of his engines, stated that it must be an instrument failure because it was a one-in-a-million chance that the readings were true. He was right in his assessment, but unfortunately, he was the one. In the United States alone there were roughly 9 million flights in 2012. So, a one-in-a-million chance could translate into nine incidents.

Sometimes, people really are at fault.

Resilience Engineering

In industrial applications, accidents in large, complex systems such as oil wells, oil refineries, chemical processing plants, electrical power systems, transportation, and medical services can have major impacts on the company and the surrounding community.

Sometimes the problems do not arise in the organization but outside it, such as when fierce storms, earthquakes, or tidal waves demolish large parts of the existing infrastructure. In either case, the question is how to design and manage these systems so that they can restore services with a minimum of disruption and damage. An important approach is *resilience engineering*, with the goal of designing systems, procedures, management, and the training of people so they are able to respond to problems as they arise. It strives to ensure that the design of all these things—the equipment, procedures, and communication both among workers and also externally to management and the public—are continually being assessed, tested, and improved.

Thus, major computer providers can deliberately cause errors in their systems to test how well the company can respond. This is done by deliberately shutting down critical facilities to ensure that the backup systems and redundancies actually work. Although it might seem dangerous to do this while the systems are online, serving real customers, the only way to test these large, complex systems is by doing so. Small tests and simulations do not carry the complexity, stress levels, and unexpected events that characterize real system failures.

As Erik Hollnagel, David Woods, and Nancy Leveson, the authors of an early influential series of books on the topic, have skillfully summarized:

> *Resilience engineering is a paradigm for safety management that focuses on how to help people cope with complexity under pressure to achieve success. It strongly contrasts with what is typical today—a paradigm of tabulating error as if it were a thing, followed by interventions to reduce this count. A resilient organisation treats safety as a core value, not a commodity that can be counted. Indeed, safety shows itself only by the events that do not happen! Rather than view past success as a reason to ramp down investments, such organisations continue to invest in anticipating the changing potential for failure because they appreciate that their knowledge of the gaps is imperfect and that their environment constantly changes. One measure of resilience is therefore the ability to create foresight—to anticipate the changing shape of risk,*

before failure and harm occurs. (Reprinted by permission of the publishers. Hollnagel, Woods, & Leveson, 2006, p. 6.)

The Paradox of Automation

Machines are getting smarter. More and more tasks are becoming fully automated. As this happens, there is a tendency to believe that many of the difficulties involved with human control will go away. Across the world, automobile accidents kill and injure tens of millions of people every year. When we finally have widespread adoption of self-driving cars, the accident and casualty rate will probably be dramatically reduced, just as automation in factories and aviation have increased efficiency while lowering both error and the rate of injury.

When automation works, it is wonderful, but when it fails, the resulting impact is usually unexpected and, as a result, dangerous. Today, automation and networked electrical generation systems have dramatically reduced the amount of time that electrical power is not available to homes and businesses. But when the electrical power grid goes down, it can affect huge sections of a country and take many days to recover. With self-driving cars, I predict that we will have fewer accidents and injuries, but that when there is an accident, it will be huge.

Automation keeps getting more and more capable. Automatic systems can take over tasks that used to be done by people, whether it is maintaining the proper temperature, automatically keeping an automobile within its assigned lane at the correct distance from the car in front, enabling airplanes to fly by themselves from takeoff to landing, or allowing ships to navigate by themselves. When the automation works, the tasks are usually done as well as or better than by people. Moreover, it saves people from the dull, dreary routine tasks, allowing more useful, productive use of time, reducing fatigue and error. But when the task gets too complex, automation tends to give up. This, of course, is precisely when it is needed the most. The paradox is that automation can take over the dull, dreary tasks, but fail with the complex ones.

When automation fails, it often does so without warning. This is a situation I have documented very thoroughly in my other books and many of my papers, as have many other people in the field of safety and automation. When the failure occurs, the human is "out of the loop." This means that the person has not been paying much attention to the operation, and it takes time for the failure to be noticed and evaluated, and then to decide how to respond.

In an airplane, when the automation fails, there is usually considerable time for the pilots to understand the situation and respond. Airplanes fly quite high: over 10 km (6 miles) above the earth, so even if the plane were to start falling, the pilots might have several minutes to respond. Moreover, pilots are extremely well trained. When automation fails in an automobile, the person might have only a fraction of a second to avoid an accident. This would be extremely difficult even for the most expert driver, and most drivers are not well trained.

In other circumstances, such as ships, there may be more time to respond, but only if the failure of the automation is noticed. In one dramatic case, the grounding of the cruise ship *Royal Majesty* in 1997, the failure lasted for several days and was only detected in the postaccident investigation, after the ship had run aground, causing several million dollars in damage. What happened? The ship's location was normally determined by the Global Positioning System (GPS), but the cable that connected the satellite antenna to the navigation system somehow had become disconnected (nobody ever discovered how). As a result, the navigation system had switched from using GPS signals to "dead reckoning," approximating the ship's location by estimating speed and direction of travel, but the design of the navigation system didn't make this apparent. As a result, as the ship traveled from Bermuda to its destination of Boston, it went too far south and went aground on Cape Cod, a peninsula jutting out of the water south of Boston. The automation had performed flawlessly for years, which increased people's trust and reliance upon it, so the normal manual checking of location or careful perusal of the display (to see the tiny letters "dr" indicating "dead reckoning" mode) were not done. This was a huge mode error failure.

Design Principles for Dealing with Error

People are flexible, versatile, and creative. Machines are rigid, precise, and relatively fixed in their operations. There is a mismatch between the two, one that can lead to enhanced capability if used properly. Think of an electronic calculator. It doesn't do mathematics like a person, but can solve problems people can't. Moreover, calculators do not make errors. So the human plus calculator is a perfect collaboration: we humans figure out what the important problems are and how to state them. Then we use calculators to compute the solutions.

Difficulties arise when we do not think of people and machines as collaborative systems, but assign whatever tasks can be automated to the machines and leave the rest to people. This ends up requiring people to behave in machine like fashion, in ways that differ from human capabilities. We expect people to monitor machines, which means keeping alert for long periods, something we are bad at. We require people to do repeated operations with the extreme precision and accuracy required by machines, again something we are not good at. When we divide up the machine and human components of a task in this way, we fail to take advantage of human strengths and capabilities but instead rely upon areas where we are genetically, biologically unsuited. Yet, when people fail, they are blamed.

What we call "human error" is often simply a human action that is inappropriate for the needs of technology. As a result, it flags a deficit in our technology. It should not be thought of as error. We should eliminate the concept of error: instead, we should realize that people can use assistance in translating their goals and plans into the appropriate form for technology.

Given the mismatch between human competencies and technological requirements, errors are inevitable. Therefore, the best designs take that fact as given and seek to minimize the opportunities for errors while also mitigating the consequences. Assume that every possible mishap will happen, so protect against them. Make actions reversible; make errors less costly. Here are key design principles:

- Put the knowledge required to operate the technology in the world. Don't require that all the knowledge must be in the head. Allow for efficient operation when people have learned all the requirements, when they are experts who can perform without the knowledge in the world, but make it possible for non-experts to use the knowledge in the world. This will also help experts who need to perform a rare, infrequently performed operation or return to the technology after a prolonged absence.
- Use the power of natural and artificial constraints: physical, logical, semantic, and cultural. Exploit the power of forcing functions and natural mappings.
- Bridge the two gulfs, the Gulf of Execution and the Gulf of Evaluation. Make things visible, both for execution and evaluation. On the execution side, provide feedforward information: make the options readily available. On the evaluation side, provide feedback: make the results of each action apparent. Make it possible to determine the system's status readily, easily, accurately, and in a form consistent with the person's goals, plans, and expectations.

We should deal with error by embracing it, by seeking to understand the causes and ensuring they do not happen again. We need to assist rather than punish or scold.