2 Questioning the Foundations of Traditional Safety Engineering

It's never what we don't know that stops us. It's what we do know that just ain't so.[1]

Paradigm changes necessarily start with questioning the basic assumptions underlying what we do today. Many beliefs about safety and why accidents occur have been widely accepted without question. This chapter examines and questions some of the most important assumptions about the cause of accidents and how to prevent them that "just ain't so." There is, of course, some truth in each of these assumptions, and many were true for the systems of the past. The real question is whether they still fit today's complex sociotechnical systems and what new assumptions need to be substituted or added.

2.1 Confusing Safety with Reliability

Assumption 1: *Safety is increased by increasing system or component reliability. If components or systems do not fail, then accidents will not occur.*

This assumption is one of the most pervasive in engineering and other fields. The problem is that it's not true. Safety and reliability are *different* properties. One does not imply nor require the other: A system can be reliable but unsafe. It can also be safe but unreliable. In some cases, these two properties even conflict, that is, making the system safer may decrease reliability and enhancing reliability may decrease safety. The confusion on this point is exemplified by the primary focus on failure events in most accident and incident analysis. Some researchers in organizational aspects of safety also make this mistake by suggesting that high *reliability* organizations will be safe [107, 175, 177, 205, 206].

1. Attributed to Will Rogers (e.g., *New York Times*, 10/7/84, p. B4), Mark Twain, and Josh Billings (*Oxford Dictionary of Quotations*, 1979, p. 49), among others.

Because this assumption about the equivalence between safety and reliability is so widely held, the distinction between these two properties needs to be carefully considered. First, let's consider accidents where none of the system components fail.

Reliable but Unsafe

In complex systems, accidents often result from interactions among components that are all satisfying their individual requirements, that is, they have *not* failed. The loss of the Mars Polar Lander was attributed to noise (spurious signals) generated when the landing legs were deployed during the spacecraft's descent to the planet surface [95]. This noise was normal and expected and did not represent a failure in the landing leg system. The onboard software interpreted these signals as an indication that landing had occurred (which the software engineers were told such signals would indicate) and shut down the descent engines prematurely, causing the spacecraft to crash into the Mars surface. The landing legs and the software performed correctly (as specified in their requirements) and reliably, but the accident occurred because the system designers did not account for all the potential interactions between landing leg deployment and the descent engine control software.

The Mars Polar Lander loss is a *component interaction accident*. Such accidents arise in the interactions among system components (electromechanical, digital, human, and social) rather than in the failure of individual components. In contrast, the other main type of accident, *a component failure accident*, results from component failures, including the possibility of multiple and cascading failures. In component failure accidents, the failures are usually treated as random phenomena. In component interaction accidents, there may be no failures and the system design errors giving rise to unsafe behavior are not random events.

A *failure* in engineering can be defined as the non-performance or inability of a component (or system) to perform its intended function. Intended function (and thus failure) is defined with respect to the component's behavioral requirements. If the behavior of a component satisfies its specified requirements (such as turning off the descent engines when a signal from the landing legs is received), even though the requirements may include behavior that is undesirable from a larger system context, that component has *not* failed.

Component failure accidents have received the most attention in engineering, but component interaction accidents are becoming more common as the complexity of our system designs increases. In the past, our designs were more intellectually manageable, and the potential interactions among components could be thoroughly planned, understood, anticipated, and guarded against [155]. In addition, thorough testing was possible and could be used to eliminate design errors before use. Modern, high-tech systems no longer have these properties, and system design errors are

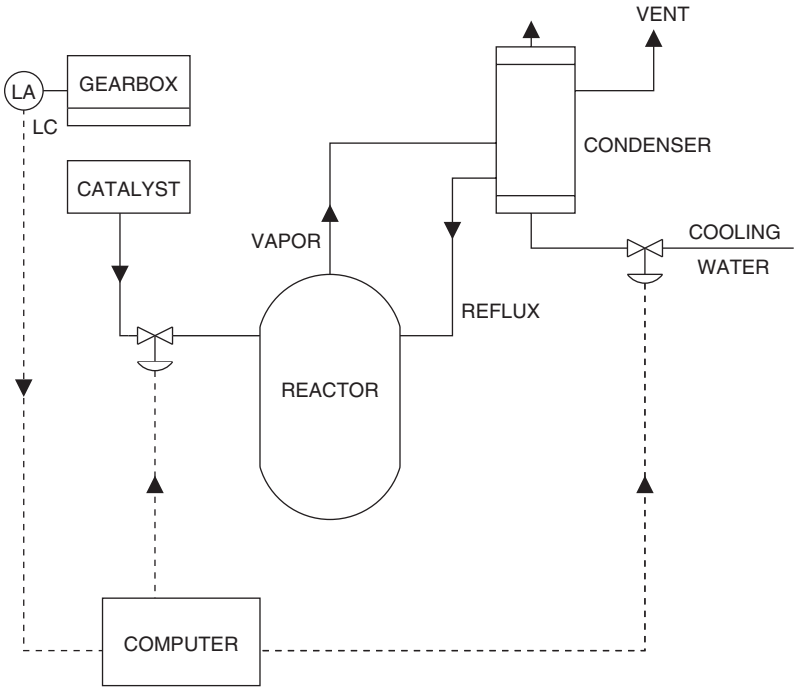


Figure 2.1
A chemical reactor design (adapted from Kletz [103, p. 6]).

increasingly the cause of major accidents, even when all the components have operated reliably—that is, the components have not failed.

Consider another example of a component interaction accident that occurred in a batch chemical reactor in England [103]. The design of this system is shown in figure 2.1. The computer was responsible for controlling the flow of catalyst into the reactor and also the flow of water into the reflux condenser to cool off the reaction. Additionally, sensor inputs to the computer were supposed to warn of any problems in various parts of the plant. The programmers were told that if a fault occurred in the plant, they were to leave all controlled variables as they were and to sound an alarm.

On one occasion, the computer received a signal indicating a low oil level in a gearbox. The computer reacted as the requirements specified: It sounded an alarm and left everything as it was. By coincidence, a catalyst had just been added to the reactor, but the computer had only started to increase the cooling-water flow to the reflux condenser; the flow was therefore kept at a low rate. The reactor overheated, the relief valve lifted, and the content of the reactor was discharged into the atmosphere.

Note that there were no component failures involved in this accident: the individual components, including the software, worked as specified, but together they created a hazardous system state. The problem was in the overall system design. Merely increasing the reliability of the individual components or protecting against their failure would not have prevented this accident because none of the components failed. Prevention required identifying and eliminating or mitigating unsafe interactions among the system components. High component reliability does not prevent component interaction accidents.

Safe but Unreliable

Accidents like the Mars Polar Lander or the British batch chemical reactor losses, where the cause lies in dysfunctional interactions of non-failing, reliable components—i.e., the problem is in the overall system design—illustrate reliable components in an unsafe system. There can also be safe systems with unreliable components if the system is designed and operated so that component failures do not create hazardous system states. Design techniques to prevent accidents are described in chapter 16 of *Safeware*. One obvious example is systems that are fail-safe, that is, they are designed to fail into a safe state.

For an example of behavior that is unreliable but safe, consider human operators. If operators do not follow the specified procedures, then they are not operating reliably. In some cases, that can lead to an accident. In other cases, it may prevent an accident when the specified procedures turn out to be unsafe under the particular circumstances existing at that time. Examples abound of operators ignoring prescribed procedures in order to prevent an accident [115, 155]. At the same time, accidents have resulted precisely because the operators *did* follow the predetermined instructions provided to them in their training, such as at Three Mile Island [115]. When the results of deviating from procedures are positive, operators are lauded, but when the results are negative, they are punished for being "unreliable." In the successful case (deviating from specified procedures averts an accident), their behavior is unreliable but safe. It satisfies the behavioral safety constraints for the system, but not individual reliability requirements with respect to following specified procedures.

It may be helpful at this point to provide some additional definitions. *Reliability* in engineering is defined as the probability that something satisfies its specified behavioral requirements over time and under given conditions—that is, it does not fail [115]. Reliability is often quantified as *mean time between failure*. Every hardware component (and most humans) can be made to "break" or fail given some set of conditions or a long enough time. The limitations in time and operating conditions in the definition are required to differentiate between (1) unreliability under the assumed operating conditions and (2) situations where no component or component design could have continued to operate.

If a driver engages the brakes of a car too late to avoid hitting the car in front, we would not say that the brakes "failed" because they did not stop the car under circumstances for which they were not designed. The brakes, in this case, were *not* unreliable. They operated reliably but the requirements for safety went beyond the capabilities of the brake design. Failure and reliability are always related to requirements and assumed operating (environmental) conditions. If there are no requirements either specified or assumed, then there can be no failure as any behavior is acceptable and no unreliability.

Safety, in contrast, is defined as the absence of accidents, where an accident is an event involving an unplanned and unacceptable loss [115]. To increase safety, the focus should be on eliminating or preventing hazards, not eliminating failures. Making all the components highly reliable will not necessarily make the system safe.

Conflicts between Safety and Reliability

At this point you may be convinced that reliable *components* are not enough for system safety. But surely, if the *system* as a whole is reliable it will be safe and vice versa, if the system is unreliable it will be unsafe. That is, reliability and safety are the same thing at the system level, aren't they? This common assumption is also untrue. A chemical plant may very reliably manufacture chemicals while occasionally (or even continually) releasing toxic materials into the surrounding environment. The plant is reliable but unsafe.

Not only are safety and reliability not the same thing, but they sometimes conflict: Increasing reliability may decrease safety and increasing safety may decrease reliability. Consider the following simple example in physical design. Increasing the working pressure to burst ratio (essentially the strength) of a tank will make the tank more reliable, that is, it will increase the mean time between failure. When a failure does occur, however, more serious damage may result because of the higher pressure at the time of the rupture.

Reliability and safety may also conflict in engineering design when a choice has to be made between retreating to a fail-safe state (and protecting people and property) versus attempting to continue to achieve the system objectives but with increased risk of an accident.

Understanding the conflicts between reliability and safety requires distinguishing between requirements and constraints. Requirements are derived from the mission or reason for the existence of the organization. The mission of the chemical plant is to produce chemicals. Constraints represent acceptable ways the system or organization can achieve the mission goals. Not exposing bystanders to toxins and not polluting the environment are constraints on the way the mission (producing chemicals) can be achieved.

While in some systems safety is part of the mission or reason for existence, such as air traffic control or healthcare, in others safety is not the mission but instead is

a constraint on how the mission can be achieved. The best way to ensure the constraints are enforced in such a system may be not to build or operate the system at all. Not building a nuclear bomb is the surest protection against accidental detonation. We may be unwilling to make that compromise, but some compromise is almost always necessary: The most effective design protections (besides not building the bomb at all) against accidental detonation also decrease the likelihood of detonation when it is required.

Not only do safety constraints sometimes conflict with mission goals, but the safety requirements may even conflict among themselves. One safety constraint on an automated train door system, for example, is that the doors must not open unless the train is stopped and properly aligned with a station platform. Another safety constraint is that the doors must open anywhere for emergency evacuation. Resolving these conflicts is one of the important steps in safety and system engineering.

Even systems with mission goals that include assuring safety, such as air traffic control (ATC), usually have other conflicting goals. ATC systems commonly have the mission to both increase system throughput and ensure safety. One way to increase throughput is to decrease safety margins by operating aircraft closer together. Keeping the aircraft separated adequately to assure acceptable risk may decrease system throughput.

There are always multiple goals and constraints for any system—the challenge in engineering design and risk management is to identify and analyze the conflicts, to make appropriate tradeoffs among the conflicting requirements and constraints, and to find ways to increase system safety without decreasing system reliability.

Safety versus Reliability at the Organizational Level

So far the discussion has focused on safety versus reliability at the physical level. But what about the social and organizational levels above the physical system? Are safety and reliability the same here as implied by High Reliability Organization (HRO) advocates who suggest that High Reliability Organizations (HROs) will be safe? The answer, again, is no [124].

Figure 2.2 shows Rasmussen's analysis of the Zeebrugge ferry mishap [167]. Some background is necessary to understand the figure. On the day the ferry capsized, the *Herald of Free Enterprise* was working the route between Dover and the Belgium port of Bruges–Zeebrugge. This route was not her normal one, and the linkspan[2] at Zeebrugge had not been designed specifically for the Spirit type of ships. The linkspan used spanned a single deck and so could not be used to load decks E and G simultaneously. The ramp could also not be raised high enough to meet the level of

2. A *linkspan* is a type of drawbridge used in moving vehicles on and off ferries or other vessels.

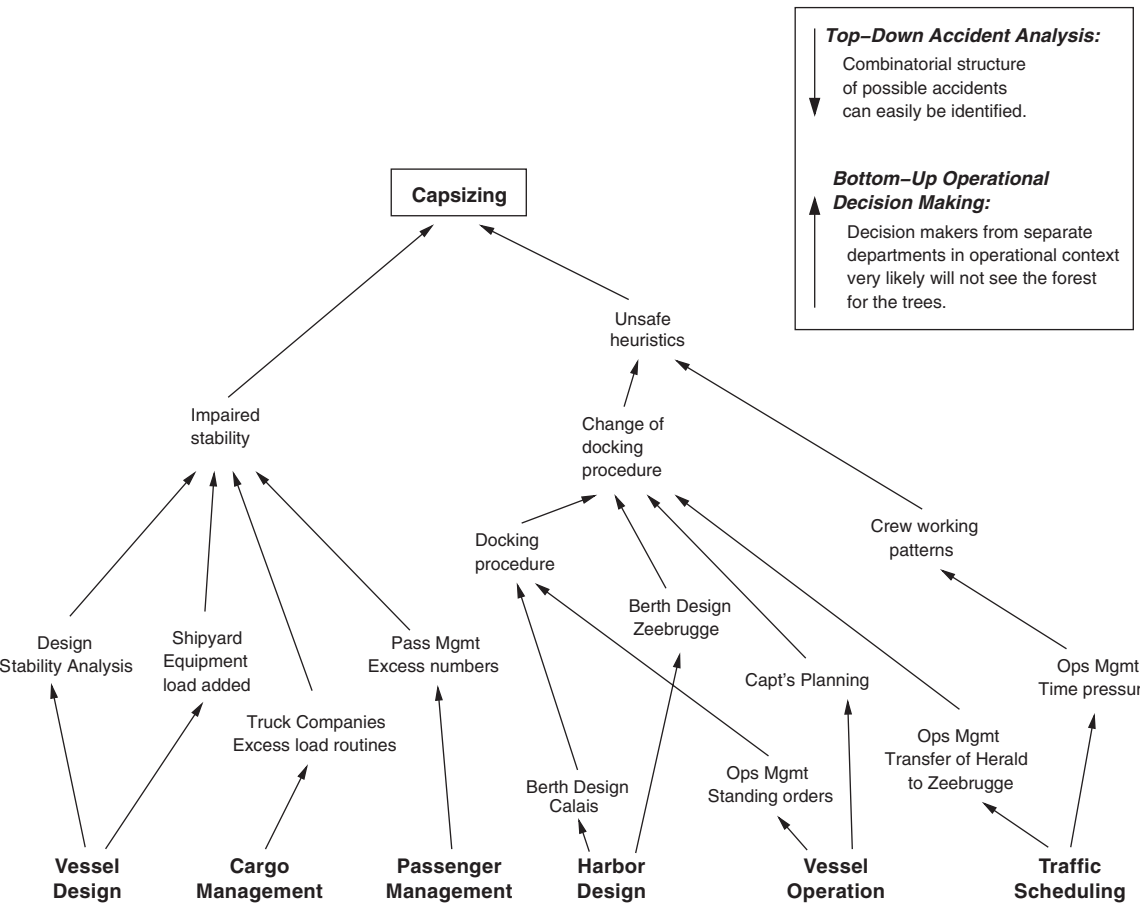


Figure 2.2
The complex interactions in the Zeebrugge accident (adapted from Rasmussen [167, p. 188]).

deck E due to the high spring tides at that time. This limitation was commonly known and was overcome by filling the forward ballast tanks to lower the ferry's bow in the water. The *Herald* was due to be modified during its refit later that year to overcome this limitation in the ship's design.

Before dropping moorings, it was normal practice for a member of the crew, the assistant boatswain, to close the ferry doors. The first officer also remained on deck to ensure they were closed before returning to the wheelhouse. On the day of the accident, in order to keep on schedule, the first officer returned to the wheelhouse before the ship dropped its moorings (which was common practice), leaving the closing of the doors to the assistant boatswain, who had taken a short break after

cleaning the car deck upon arrival at Zeebrugge. He had returned to his cabin and was still asleep when the ship left the dock. The captain could only assume that the doors had been closed because he could not see them from the wheelhouse due to their construction, and there was no indicator light in the wheelhouse to show door position. Why nobody else closed the door is unexplained in the accident report.

Other factors also contributed to the loss. One was the depth of the water: if the ship's speed had been below 18 knots (33 km/h) and the ship had not been in shallow water, it was speculated in the accident report that the people on the car deck would probably have had time to notice the bow doors were open and close them [187]. But open bow doors were not alone enough to cause the final capsizing. A few years earlier, one of the *Herald*'s sister ships sailed from Dover to Zeebrugge with the bow doors open and made it to her destination without incident.

Almost all ships are divided into watertight compartments below the waterline so that in the event of flooding, the water will be confined to one compartment, keeping the ship afloat. The *Herald*'s design had an open car deck with no dividers, allowing vehicles to drive in and out easily, but this design allowed water to flood the car deck. As the ferry turned, the water on the car deck moved to one side and the vessel capsized. One hundred and ninety three passengers and crew were killed.

In this accident, those making decisions about vessel design, harbor design, cargo management, passenger management, traffic scheduling, and vessel operation were unaware of the impact (side effects) of their decisions on the others and the overall impact on the process leading to the ferry accident. Each operated "reliably" in terms of making decisions based on the information they had.

Bottom-up decentralized decision making can lead—and has led—to major accidents in complex sociotechnical systems. Each local decision may be "correct" in the limited context in which it was made but lead to an accident when the independent decisions and organizational behaviors interact in dysfunctional ways.

Safety is a system property, not a component property, and must be controlled at the system level, not the component level. We return to this topic in chapter 3.

Assumption 1 is clearly untrue. A new assumption needs to be substituted:

New Assumption 1: *High reliability is neither necessary nor sufficient for safety.*

Building safer systems requires going beyond the usual focus on component failure and reliability to focus on system hazards and eliminating or reducing their occurrence. This fact has important implications for analyzing and designing for safety. Bottom-up reliability engineering analysis techniques, such as failure modes and effects analysis (FMEA), are not appropriate for safety analysis. Even top-down techniques, such as fault trees, if they focus on component failure, are not adequate. Something else is needed.

2.2 Modeling Accident Causation as Event Chains

Assumption 2: *Accidents are caused by chains of directly related events. We can understand accidents and assess risk by looking at the chain of events leading to the loss.*

Some of the most important assumptions in safety lie in our models of how the world works. Models are important because they provide a means for understanding phenomena like accidents or potentially hazardous system behavior and for recording that understanding in a way that can be communicated to others.

A particular type of model, an *accident causality model* (or *accident model* for short) underlies all efforts to engineer for safety. Our accident models provide the foundation for (1) investigating and analyzing the cause of accidents, (2) designing to prevent future losses, and (3) assessing the risk associated with using the systems and products we create. Accident models explain why accidents occur, and they determine the approaches we take to prevent them. While you might not be consciously aware you are using a model when engaged in these activities, some (perhaps subconscious) model of the phenomenon is always part of the process.

All models are abstractions; they simplify the thing being modeled by abstracting away what are assumed to be irrelevant details and focusing on the features of the phenomenon that are judged to be the most relevant. Selecting some factors as relevant and others as irrelevant is, in most cases, arbitrary and entirely the choice of the modeler. That choice, however, is critical in determining the usefulness and accuracy of the model in predicting future events.

An underlying assumption of all accident models is that there are common patterns in accidents and that they are not simply random events. Accident models impose patterns on accidents and influence the factors considered in any safety analysis. Because the accident model influences what cause(s) is ascribed to an accident, the countermeasures taken to prevent future accidents, and the evaluation of the risk in operating a system, the power and features of the accident model used will greatly affect our ability to identify and control hazards and thus prevent accidents.

The earliest formal accident models came from *industrial safety* (sometimes called *occupational safety*) and reflect the factors inherent in protecting workers from injury or illness. Later, these same models or variants of them were applied to the engineering and operation of complex technical and social systems. At the beginning, the focus in industrial accident prevention was on unsafe conditions, such as open blades and unprotected belts. While this emphasis on preventing unsafe conditions was very successful in reducing workplace injuries, the decrease naturally started to slow down as the most obvious hazards were eliminated. The emphasis

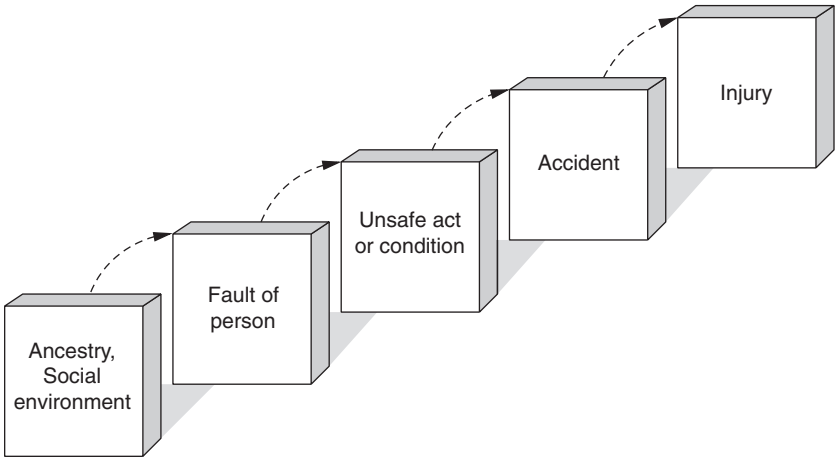


Figure 2.3
Heinrich's Domino Model of Accidents.

then shifted to unsafe acts: Accidents began to be regarded as someone's fault rather than as an event that could have been prevented by some change in the plant or product.

Heinrich's Domino Model, published in 1931, was one of the first published general accident models and was very influential in shifting the emphasis in safety to human error. Heinrich compared the general sequence of accidents to five dominoes standing on end in a line (figure 2.3). When the first domino falls, it automatically knocks down its neighbor and so on until the injury occurs. In any accident sequence, according to this model, ancestry or social environment leads to a fault of a person, which is the proximate reason for an unsafe act or condition (mechanical or physical), which results in an accident, which leads to an injury. In 1976, Bird and Loftus extended the basic Domino Model to include management decisions as a factor in accidents:

1. Lack of control by management, permitting
2. Basic causes (personal and job factors) that lead to
3. Immediate causes (substandard practices/conditions/errors), which are the proximate cause of
4. An accident or incident, which results in
5. A loss

In the same year, Adams suggested a different management-augmented model that included:

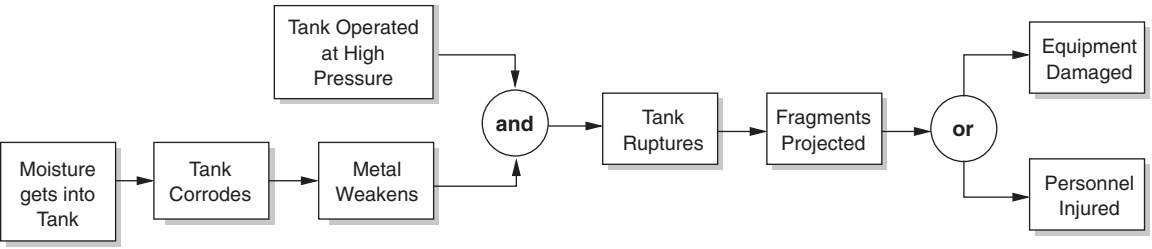


Figure 2.4
A model of the chain of events leading to the rupture of a pressurized tank (adapted from Hammer [79]). Moisture leads to corrosion, which causes weakened metal, which together with high operating pressures causes the tank to rupture, resulting in fragments being projected, and finally leading to personnel injury and/or equipment failure.

1. Management structure (objectives, organization, and operations)
2. Operational errors (management or supervisory behavior)
3. Tactical errors (caused by employee behavior and work conditions)
4. Accident or incident
5. Injury or damage to persons or property

Reason reinvented the Domino Model twenty years later in what he called the Swiss Cheese model, with layers of Swiss cheese substituted for dominos and the layers or dominos labeled as layers of defense[3] that have failed [172, 173].

The basic Domino Model is inadequate for complex systems and other models were developed (see *Safeware* [115], chapter 10), but the assumption that there is a single or *root cause* of an accident unfortunately persists as does the idea of dominos (or layers of Swiss cheese) and chains of failures, each directly causing or leading to the next one in the chain. It also lives on in the emphasis on human error in identifying accident causes.

The most common accident models today explain accidents in terms of multiple events sequenced as a forward chain over time. The events included almost always involve some type of failure" event or human error, or they are energy related (for example, an explosion). The chains may be branching (as in fault trees) or there may be multiple chains synchronized by time or common events. Lots of notations have been developed to represent the events in a graphical form, but the underlying model is the same. Figure 2.4 shows an example for the rupture of a pressurized tank.

3. Designing layers of defense is a common safety design approach used primarily in the process industry, particularly for nuclear power. Different design approaches are commonly used in other industries.

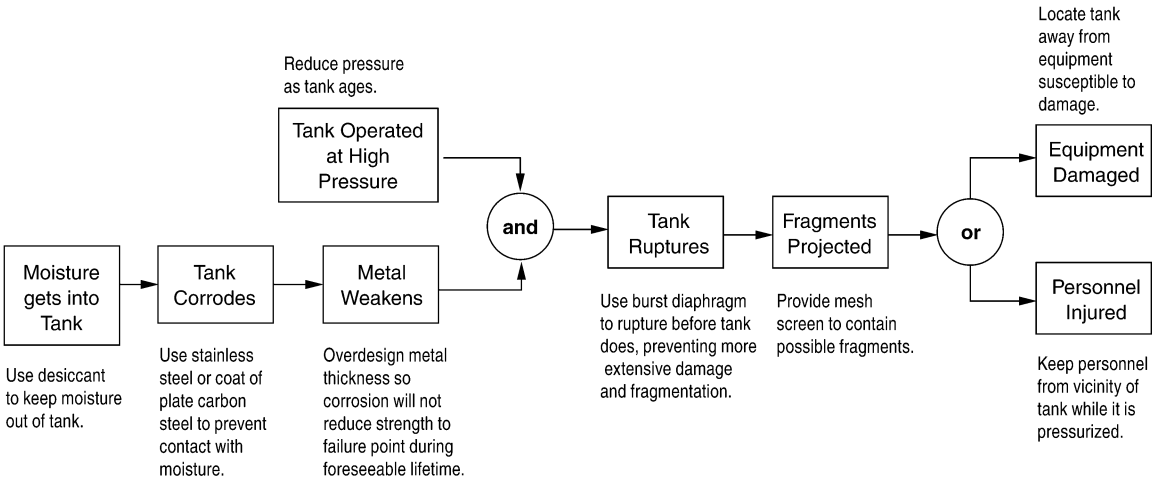


Figure 2.5
The pressurized tank rupture event chain along with measures that could be taken to "break" the chain by preventing individual events in it.

The use of event-chain models of causation has important implications for the way engineers design for safety. If an accident is caused by a chain of events, then the most obvious preventive measure is to break the chain before the loss occurs. Because the most common events considered in these models are component failures, preventive measures tend to be focused on preventing failure events—increasing component integrity or introducing redundancy to reduce the likelihood of the event occurring. If corrosion can be prevented in the tank rupture accident, for example, then the tank rupture is averted.

Figure 2.5 is annotated with mitigation measures designed to break the chain. These mitigation measures are examples of the most common design techniques based on event-chain models of accidents, such as barriers (for example, preventing the contact of moisture with the metal used in the tank by coating it with plate carbon steel or providing mesh screens to contain fragments), interlocks (using a burst diaphragm), overdesign (increasing the metal thickness), and operational procedures (reducing the amount of pressure as the tank ages).

For this simple example involving only physical failures, designing to prevent such failures works well. But even this simple example omits any consideration of factors indirectly related to the events in the chain. An example of a possible indirect or systemic example is competitive or financial pressures to increase efficiency that could lead to not following the plan to reduce the operating pressure as the tank ages. A second factor might be changes over time to the plant design that require workers to spend time near the tank while it is pressurized.

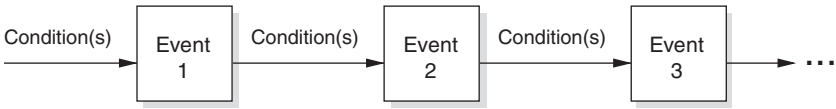


Figure 2.6
Conditions cause events, which lead to new conditions, which cause further events . . .

Formal and informal notations for representing the event chain may contain only the events or they may also contain the conditions that led to the events. Events create conditions that, along with existing conditions, lead to events that create new conditions, and so on (figure 2.6). The *tank corrodes* event leads to a *corrosion exists in tank* condition, which leads to a *metal weakens* event, which leads to a *weakened metal* condition, and so forth.

The difference between events and conditions is that events are limited in time, while conditions persist until some event occurs that results in new or changed conditions. For example, the three conditions that must exist before a flammable mixture will explode (the event) are the flammable gases or vapors themselves, air, and a source of ignition. Any one or two of these may exist for a period of time before the other(s) occurs and leads to the explosion. An event (the explosion) creates new conditions, such as uncontrolled energy or toxic chemicals in the air.

Causality models based on event chains (or dominos or layers of Swiss cheese) are simple and therefore appealing. But they are too simple and do not include what is needed to understand why accidents occur and how to prevent them. Some important limitations include requiring direct causality relationships, subjectivity in selecting the events to include, subjectivity in identifying chaining conditions, and exclusion of systemic factors.

2.2.1 Direct Causality

The causal relationships between the events in event chain models (or between dominoes or Swiss cheese slices) are required to be direct and linear, representing the notion that the preceding event must have occurred and the linking conditions must have been present for the subsequent event to occur: if event A had not occurred then the following event B would not have occurred. As such, event chain models encourage limited notions of linear causality, and it is difficult or impossible to incorporate nonlinear relationships. Consider the statement "Smoking causes lung cancer." Such a statement would not be allowed in the event-chain model of causality because there is no direct relationship between the two. Many smokers do not get lung cancer, and some people who get lung cancer are not smokers. It is widely accepted, however, that there is some relationship between the two, although it may be quite complex and nonlinear.

In addition to limitations in the types of causality considered, the causal factors identified using event-chain models depend on the events that are considered and on the selection of the conditions that link the events. Other than the physical events immediately preceding or directly involved in the loss, however, the choice of events to include is subjective and the conditions selected to explain the events is even more so. Each of these two limitations is considered in turn.

2.2.2 Subjectivity in Selecting Events

The selection of events to include in an event chain is dependent on the stopping rule used to determine how far back the sequence of explanatory events goes. Although the first event in the chain is often labeled the *initiating event* or *root cause*, the selection of an initiating event is arbitrary and previous events and conditions could always be added.

Sometimes the initiating event is selected (the backward chaining stops) because it represents a type of event that is familiar and thus acceptable as an explanation for the accident or it is a deviation from a standard [166]. In other cases, the initiating event or root cause is chosen because it is the first event in the backward chain for which it is felt that something can be done for correction.[4]

The backward chaining may also stop because the causal path disappears due to lack of information. Rasmussen suggests that a practical explanation for why actions by operators actively involved in the dynamic flow of events are so often identified as the cause of an accident is the difficulty in continuing the backtracking "through" a human [166].

A final reason why a "root cause" may be selected is that it is politically acceptable as the identified cause. Other events or explanations may be excluded or not examined in depth because they raise issues that are embarrassing to the organization or its contractors or are politically unacceptable.

The accident report on a friendly fire shootdown of a U.S. Army helicopter over the Iraqi no-fly zone in 1994, for example, describes the chain of events leading to the shootdown. Included in these events is the fact that the helicopter pilots did not change to the radio frequency required in the no-fly zone when they entered it (they stayed on the enroute frequency). Stopping at this event in the chain (which the official report does), it appears that the helicopter pilots were partially at fault for the loss by not following radio procedures. An independent account of the accident [159], however, notes that the U.S. commander of the operation had made

4. As an example, a NASA Procedures and Guidelines document (NPG 8621 Draft 1) defines a root cause as: "Along a chain of events leading to an mishap, the first causal action or failure to act that could have been controlled systematically either by policy/practice/procedure or individual adherence to policy/practice/procedure."

an exception about the radio frequency to be used by the helicopters in order to mitigate a different safety concern (see chapter 5), and therefore the pilots were simply following orders when they did not switch to the "required" frequency. The command to the helicopter pilots not to follow official radio procedures is not included in the chain of events provided in the official government accident report, but it suggests a very different understanding of the role of the helicopter pilots in the loss.

In addition to a *root* cause or causes, some events or conditions may be identified as *proximate* or *direct* causes while others are labeled as *contributory*. There is no more basis for this distinction than the selection of a root cause.

Making such distinctions between causes or limiting the factors considered can be a hindrance in learning from and preventing future accidents. Consider the following aircraft examples.

In the crash of an American Airlines DC-10 at Chicago's O'Hare Airport in 1979, the U.S. National Transportation Safety Board (NTSB) blamed only a "maintenance-induced crack," and not also a design error that allowed the slats to retract if the wing was punctured. Because of this omission, McDonnell Douglas was not required to change the design, leading to future accidents related to the same design flaw [155].

Similar omissions of causal factors in aircraft accidents have occurred more recently. One example is the crash of a China Airlines A300 on April 26, 1994, while approaching the Nagoya, Japan, airport. One of the factors involved in the accident was the design of the flight control computer software. Previous incidents with the same type of aircraft had led to a Service Bulletin being issued for a modification of the two flight control computers to fix the problem. But because the computer problem had not been labeled a "cause" of the previous incidents (for perhaps at least partially political reasons), the modification was labeled *recommended* rather than *mandatory*. China Airlines concluded, as a result, that the implementation of the changes to the computers was not urgent and decided to delay modification until the next time the flight computers on the plane needed repair [4]. Because of that delay, 264 passengers and crew died.

In another DC-10 saga, explosive decompression played a critical role in a near miss over Windsor, Ontario. An American Airlines DC-10 lost part of its passenger floor, and thus all of the control cables that ran through it, when a cargo door opened in flight in June 1972. Thanks to the extraordinary skill and poise of the pilot, Bryce McCormick, the plane landed safely. In a remarkable coincidence, McCormick had trained himself to fly the plane using only the engines because he had been concerned about a decompression-caused collapse of the floor. After this close call, McCormick recommended that every DC-10 pilot be informed of the consequences of explosive decompression and trained in the flying techniques that he and his crew

had used to save their passengers and aircraft. FAA investigators, the National Transportation Safety Board, and engineers at a subcontractor to McDonnell Douglas that designed the fuselage of the plane, all recommended changes in the design of the aircraft. Instead, McDonnell Douglas attributed the Windsor incident totally to human error on the part of the baggage handler responsible for closing the cargo compartment door (a convenient event in the event chain) and not to any error on the part of their designers or engineers and decided all they had to do was to come up with a fix that would prevent baggage handlers from forcing the door.

One of the discoveries after the Windsor incident was that the door could be improperly closed but the external signs, such as the position of the external handle, made it appear to be closed properly. In addition, this incident proved that the cockpit warning system could fail, and the crew would then not know that the plane was taking off without a properly closed door:

> The aviation industry does not normally receive such manifest warnings of basic design flaws in an aircraft without cost to human life. Windsor deserved to be celebrated as an exceptional case when every life was saved through a combination of crew skill and the sheer luck that the plane was so lightly loaded. If there had been more passengers and thus more weight, damage to the control cables would undoubtedly have been more severe, and it is highly questionable if any amount of skill could have saved the plane [61].

Almost two years later, in March 1974, a fully loaded Turkish Airlines DC-10 crashed near Paris, resulting in 346 deaths—one of the worst accidents in aviation history. Once again, the cargo door had opened in flight, causing the cabin floor to collapse, severing the flight control cables. Immediately after the accident, Sanford McDonnell stated the official McDonnell-Douglas position that once again placed the blame on the baggage handler and the ground crew. This time, however, the FAA finally ordered modifications to all DC-10s that eliminated the hazard. In addition, an FAA regulation issued in July 1975 required all wide-bodied jets to be able to tolerate a hole in the fuselage of twenty square feet. By labeling the root cause in the event chain as baggage handler error and attempting only to eliminate that event or link in the chain rather than the basic engineering design flaws, fixes that could have prevented the Paris crash were not made.

Until we do a better job of identifying causal factors in accidents, we will continue to have unnecessary repetition of incidents and accidents.

2.2.3 Subjectivity in Selecting the Chaining Conditions

In addition to subjectivity in selecting the events and the root cause event, the links between the events that are chosen to explain them are subjective and subject to bias. Leplat notes that the links are justified by knowledge or rules of different types, including physical and organizational knowledge. The same event can give rise to different types of links according to the mental representations the analyst has of

the production of this event. When several types of rules are possible, the analyst will apply those that agree with his or her mental model of the situation [111].

Consider, for example, the loss of an American Airlines B757 near Cali, Colombia, in 1995 [2]. Two significant events in this loss were

(1) Pilot asks for clearance to take the rozo approach

followed later by

(2) Pilot types R into the FMS.[5]

In fact, the pilot should have typed the four letters ROZO instead of *R*—the latter was the symbol for a different radio beacon (called ROMEO) near Bogota. As a result, the aircraft incorrectly turned toward mountainous terrain. While these events are noncontroversial, the link between the two events could be explained by any of the following:

- *Pilot Error:* In the rush to start the descent, the pilot executed a change of course without verifying its effect on the flight path.
- *Crew Procedure Error:* In the rush to start the descent, the captain entered the name of the waypoint without normal verification from the other pilot.
- *Approach Chart and FMS Inconsistencies:* The identifier used to identify rozo on the approach chart (R) did not match the identifier used to call up rozo in the FMS.
- *FMS Design Deficiency:* The FMS did not provide the pilot with feedback that choosing the first identifier listed on the display was not the closest beacon having that identifier.
- *American Airlines Training Deficiency:* The pilots flying into South America were not warned about duplicate beacon identifiers nor adequately trained on the logic and priorities used in the FMS on the aircraft.
- *Manufacturer Deficiency:* Jeppesen-Sanderson did not inform airlines operating FMS-equipped aircraft of the differences between navigation information provided by Jeppesen-Sanderson Flight Management System navigation databases and Jeppesen-Sanderson approach charts or the logic and priorities employed in the display of electronic FMS navigation information.
- *International Standards Deficiency:* No single worldwide standard provides unified criteria for the providers of electronic navigation databases used in Flight Management Systems.

5. An FMS is an automated flight management system that assists the pilots in various ways. In this case, it was being used to provide navigation information.

The selection of the linking condition (or events) will greatly influence the cause ascribed to the accident yet in the example all are plausible and each could serve as an explanation of the event sequence. The choice may reflect more on the person or group making the selection than on the accident itself. In fact, understanding this accident and learning enough from it to prevent future accidents requires identifying *all* of these factors to explain the incorrect input: The accident model used should encourage and guide a comprehensive analysis at multiple technical and social system levels.

2.2.4 Discounting Systemic Factors

The problem with event chain models is not simply that the selection of the events to include and the labeling of some of them as causes are arbitrary or that the selection of which conditions to include is also arbitrary and usually incomplete. Even more important is that viewing accidents as chains of events and conditions may limit understanding and learning from the loss and omit causal factors that cannot be included in an event chain.

Event chains developed to explain an accident usually concentrate on the proximate events immediately preceding the loss. But the foundation for an accident is often laid years before. One event simply triggers the loss, but if that event had not happened, another one would have led to a loss. The Bhopal disaster provides a good example.

The release of methyl isocyanate (MIC) from the Union Carbide chemical plant in Bhopal, India, in December 1984 has been called the worst industrial accident in history: Conservative estimates point to 2,000 fatalities, 10,000 permanent disabilities (including blindness), and 200,000 injuries [38]. The Indian government blamed the accident on human error—the improper cleaning of a pipe at the plant. A relatively new worker was assigned to wash out some pipes and filters, which were clogged. MIC produces large amounts of heat when in contact with water, and the worker properly closed the valves to isolate the MIC tanks from the pipes and filters being washed. Nobody, however, inserted a required safety disk (called a *slip blind*) to back up the valves in case they leaked [12].

A chain of events describing the accident mechanism for Bhopal might include:

E1 Worker washes pipes without inserting a slip blind.

E2 Water leaks into MIC tank.

E3 Explosion occurs.

E4 Relief valve opens.

E5 MIC vented into air.

E6 Wind carries MIC into populated area around plant.

Both Union Carbide and the Indian government blamed the worker washing the pipes for the accident.[6] A different operator error might be identified as the root cause (initiating event) if the chain is followed back farther. The worker who had been assigned the task of washing the pipes reportedly knew that the valves leaked, but he did not check to see whether the pipe was properly isolated because, he said, it was not his job to do so. Inserting the safety disks was the job of the maintenance department, but the maintenance sheet contained no instruction to insert this disk. The pipe-washing operation should have been supervised by the second shift supervisor, but that position had been eliminated in a cost-cutting effort. So the root cause might instead have been assigned to the person responsible for inserting the slip blind or to the lack of a second shift supervisor.

But the selection of a stopping point and the specific operator action to label as the root cause—and operator actions are almost always selected as root causes—is not the real problem here. The problem is the oversimplification implicit in using a chain of events to understand why this accident occurred. Given the design and operating conditions of the plant, an accident was waiting to happen:

> However [water] got in, it would not have caused the severe explosion had the refrigeration unit not been disconnected and drained of freon, or had the gauges been properly working and monitored, or had various steps been taken at the first smell of MIC instead of being put off until after the tea break, or had the scrubber been in service, or had the water sprays been designed to go high enough to douse the emissions, or had the flare tower been working and been of sufficient capacity to handle a large excursion. [156, p. 349]

It is not uncommon for a company to turn off passive safety devices, such as refrigeration units, to save money. The operating manual specified that the refrigeration unit *must* be operating whenever MIC was in the system: The chemical has to be maintained at a temperature no higher than 5° Celsius to avoid uncontrolled reactions. A high temperature alarm was to sound if the MIC reached 11°. The refrigeration unit was turned off, however, to save money, and the MIC was usually stored at nearly 20°. The plant management adjusted the threshold of the alarm, accordingly, from 11° to 20° and logging of tank temperatures was halted, thus eliminating the possibility of an early warning of rising temperatures.

Gauges at plants are frequently out of service [23]. At the Bhopal facility, there were few alarms or interlock devices in critical locations that might have warned operators of abnormal conditions—a system design deficiency.

6. Union Carbide lawyers argued that the introduction of water into the MIC tank was an act of sabotage rather than a maintenance worker's mistake. While this differing interpretation of the initiating event has important implications with respect to legal liability, it makes no difference in the argument presented here regarding the limitations of event-chain models of accidents or even, as will be seen, understanding why this accident occurred.

Other protection devices at the plant had inadequate design thresholds. The vent scrubber, had it worked, was designed to neutralize only small quantities of gas at fairly low pressures and temperatures: The pressure of the escaping gas during the accident exceeded the scrubber's design by nearly two and a half times, and the temperature of the escaping gas was at least 80° Celsius more than the scrubber could handle. Similarly, the flare tower (which was supposed to burn off released vapor) was totally inadequate to deal with the estimated 40 tons of MIC that escaped during the accident. In addition, the MIC was vented from the vent stack 108 feet above the ground, well above the height of the water curtain intended to knock down the gas: The water curtain reached only 40 to 50 feet above the ground. The water jets could reach as high as 115 feet, but only if operated individually.

Leaks were routine occurrences and the reasons for them were seldom investigated: Problems were either fixed without further examination or were ignored. A safety audit two years earlier by a team from Union Carbide had noted many safety problems at the plant, including several involved in the accident, such as filter-cleaning operations without using slip blinds, leaking valves, the possibility of contaminating the tank with material from the vent gas scrubber, and bad pressure gauges. The safety auditors had recommended increasing the capability of the water curtain and had pointed out that the alarm at the flare tower from which the MIC leaked was nonoperational, and thus any leak could go unnoticed for a long time. None of the recommended changes were made [23]. There is debate about whether the audit information was fully shared with the Union Carbide India subsidiary and about who was responsible for making sure changes were made. In any event, there was no follow-up to make sure that the problems identified in the audit had been corrected.

A year before the accident, the chemical engineer managing the MIC plant resigned because he disapproved of falling safety standards, and still no changes were made. He was replaced by an electrical engineer. Measures for dealing with a chemical release once it occurred were no better. Alarms at the plant sounded so often (the siren went off twenty to thirty times a week for various purposes) that an actual alert could not be distinguished from routine events or practice alerts. Ironically, the warning siren was not turned on until two hours after the MIC leak was detected (and after almost all the injuries had occurred) and then was turned off after only five minutes—which was company policy [12]. Moreover, the numerous practice alerts did not seem to be effective in preparing for an emergency: When the danger during the release became known, many employees ran from the contaminated areas of the plant, totally ignoring the buses that were sitting idle ready to evacuate workers and nearby residents. Plant workers had only a bare minimum of emergency equipment—a shortage of oxygen masks, for example, was discovered after the accident started—and they had almost no knowledge or training about how to handle nonroutine events.

The police were not notified when the chemical release began. In fact, when called by police and reporters, plant spokesmen first denied the accident and then claimed that MIC was not dangerous. Nor was the surrounding community warned of the dangers, before or during the release, or informed of the simple precautions that could have saved them from lethal exposure, such as putting a wet cloth over their face and closing their eyes. If the community had been alerted and provided with this simple information, many (if not most) lives would have been saved and injuries prevented [106].

Some of the reasons why the poor conditions in the plant were allowed to persist are financial. Demand for MIC had dropped sharply after 1981, leading to reductions in production and pressure on the company to cut costs. The plant was operating at less than half capacity when the accident occurred. Union Carbide put pressure on the Indian management to reduce losses, but gave no specific details on how to achieve the reductions. In response, the maintenance and operating personnel were cut in half. Maintenance procedures were severely cut back and the shift relieving system was suspended—if no replacement showed up at the end of the shift, the following shift went unmanned. The person responsible for inserting the slip blind in the pipe had not showed up for his shift. Top management justified the cuts as merely reducing avoidable and wasteful expenditures without affecting overall safety.

As the plant lost money, many of the skilled workers left for more secure jobs. They either were not replaced or were replaced by unskilled workers. When the plant was first built, operators and technicians had the equivalent of two years of college education in chemistry or chemical engineering. In addition, Union Carbide provided them with six months training. When the plant began to lose money, educational standards and staffing levels were reportedly reduced. In the past, UC flew plant personnel to West Virginia for intensive training and had teams of U.S. engineers make regular on-site safety inspections. But by 1982, financial pressures led UC to give up direct supervision of safety at the plant, even though it retained general financial and technical control. No American advisors were resident at Bhopal after 1982.

Management and labor problems followed the financial losses. Morale at the plant was low. "There was widespread belief among employees that the management had taken drastic and imprudent measures to cut costs and that attention to details that ensure safe operation were absent" [127].

These are only a few of the factors involved in this catastrophe, which also include other technical and human errors within the plant, design errors, management negligence, regulatory deficiencies on the part of the U.S. and Indian governments, and general agricultural and technology transfer policies related to the reason they were making such a dangerous chemical in India in the first place. Any one of these perspectives or "causes" is inadequate by itself to understand the accident and to

prevent future ones. In particular, identifying only operator error or sabotage as the root cause of the accident ignores most of the opportunities for the prevention of similar accidents in the future. Many of the systemic causal factors are only indirectly related to the proximate events and conditions preceding the loss.

When all the factors, including indirect and systemic ones, are considered, it becomes clear that the maintenance worker was, in fact, only a minor and somewhat irrelevant player in the loss. Instead, degradation in the safety margin occurred over time and without any particular single decision to do so but simply as a series of decisions that moved the plant slowly toward a situation where any slight error would lead to a major accident. Given the overall state of the Bhopal Union Carbide plant and its operation, if the action of inserting the slip disk had not been left out of the pipe washing operation that December day in 1984, something else would have triggered an accident. In fact, a similar leak had occurred the year before, but did not have the same catastrophic consequences and the true root causes of that incident were neither identified nor fixed.

To label one event (such as a maintenance worker leaving out the slip disk) or even several events as the root cause or the start of an event chain leading to the Bhopal accident is misleading at best. Rasmussen writes:

> The stage for an accidental course of events very likely is prepared through time by the normal efforts of many actors in their respective daily work context, responding to the standing request to be more productive and less costly. Ultimately, a quite normal variation in somebody's behavior can then release an accident. Had this "root cause" been avoided by some additional safety measure, the accident would very likely be released by another cause at another point in time. In other words, an explanation of the accident in terms of events, acts, and errors is not very useful for design of improved systems [167].

In general, event-based models are poor at representing systemic accident factors such as structural deficiencies in the organization, management decision making, and flaws in the safety culture of the company or industry. An accident model should encourage a broad view of accident mechanisms that expands the investigation beyond the proximate events: A narrow focus on technological components and pure engineering activities or a similar narrow focus on operator errors may lead to ignoring some of the most important factors in terms of preventing future accidents. The accident model used to explain why the accident occurred should not only encourage the inclusion of all the causal factors but should provide guidance in identifying these factors.

2.2.5 Including Systems Factors in Accident Models

Large-scale engineered systems are more than just a collection of technological artifacts: They are a reflection of the structure, management, procedures, and culture of the engineering organization that created them. They are usually also a reflection

of the society in which they were created. Ralph Miles Jr., in describing the basic concepts of systems theory, notes,

> Underlying every technology is at least one basic science, although the technology may be well developed long before the science emerges. Overlying every technical or civil system is a social system that provides purpose, goals, and decision criteria. [137, p. 1]

Effectively preventing accidents in complex systems requires using accident models that include that social system as well as the technology and its underlying science. Without understanding the purpose, goals, and decision criteria used to construct and operate systems, it is not possible to completely understand and most effectively prevent accidents.

Awareness of the importance of social and organizational aspects of safety goes back to the early days of System Safety.[7] In 1968, Jerome Lederer, then the director of the NASA Manned Flight Safety Program for Apollo, wrote:

> System safety covers the total spectrum of risk management. It goes *beyond the hardware* and associated procedures of system safety engineering. It involves: attitudes and motivation of designers and production people, employee/management rapport, the relation of industrial associations among themselves and with government, human factors in supervision and quality control, documentation on the interfaces of industrial and public safety with design and operations, the interest and attitudes of top management, the effects of the legal system on accident investigations and exchange of information, the certification of critical workers, political considerations, resources, public sentiment and many other non-technical but vital influences on the attainment of an acceptable level of risk control. These non-technical aspects of system safety cannot be ignored. [109]

Too often, however, these non-technical aspects *are* ignored.

At least three types of factors need to be considered in accident causation. The first is the proximate event chain, which for the *Herald of Free Enterprise* includes the assistant boatswain's not closing the doors and the return of the first officer to the wheelhouse prematurely. Note that there was a redundant design here, with the first officer checking the work of the assistant boatswain, but it did not prevent the accident, as is often the case with redundancy [115, 155].

The second type of information includes the conditions that allowed the events to occur: the high spring tides, the inadequate design of the ferry loading ramp for this harbor, and the desire of the first officer to stay on schedule (thus leaving the car deck before the doors were closed). All of these conditions can be directly mapped to the events.

---

7. When this term is capitalized in this book, it denotes the specific form of safety engineering developed originally by the Defense Department and its contractors for the early ICBM systems and defined by MIL-STD-882. System safety (uncapitalized) or safety engineering denotes all the approaches to engineering for safety.

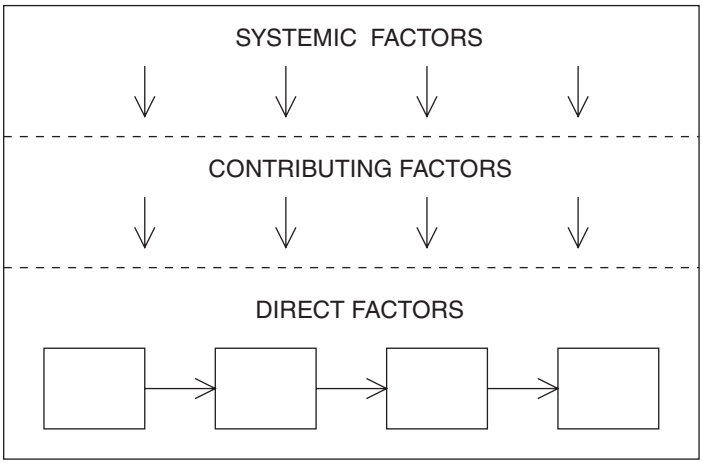


Figure 2.7
Johnson's three-level model of accidents.

The third set of causal factors is only indirectly related to the events and conditions, but these indirect factors are critical in fully understanding why the accident occurred and thus how to prevent future accidents. In this case, the systemic factors include the owner of the ferry (Townsend Thoresen) needing ships that were designed to permit fast loading and unloading and quick acceleration in order to remain competitive in the ferry business, and pressure by company management on the captain and first officer to strictly adhere to schedules, also related to competitive factors.

Several attempts have been made to graft systemic factors onto event models, but all have important limitations. The most common approach has been to add hierarchical levels above the event chain. In the seventies, Johnson proposed a model and sequencing method that described accidents as chains of direct events and causal factors arising from contributory factors, which in turn arise from systemic factors (figure 2.7) [93].

Johnson also tried to put management factors into fault trees (a technique called MORT, or Management Oversight Risk Tree), but ended up simply providing a general checklist for auditing management practices. While such a checklist can be very useful, it presupposes that every error can be predefined and put into a checklist form. The checklist is comprised of a set of questions that should be asked during an accident investigation. Examples of the questions from a DOE MORT User's Manual are: Was there sufficient training to update and improve needed supervisory skills? Did the supervisors have their own technical staff or access to such individuals? Was there technical support of the right discipline(s) sufficient for the needs of

supervisory programs and review functions? Were there established methods for measuring performance that permitted the effectiveness of supervisory programs to be evaluated? Was a maintenance plan provided before startup? Was all relevant information provided to planners and managers? Was it used? Was concern for safety displayed by vigorous, visible personal action by top executives? And so forth.

Johnson originally provided hundreds of such questions, and additions have been made to his checklist since Johnson created it in the 1970s so it is now even larger. The use of the MORT checklist is feasible because the items are so general, but that same generality also limits its usefulness. Something more effective than checklists is needed.

The most sophisticated of the hierarchical add-ons to event chains is Rasmussen and Svedung's model of the sociotechnical system involved in risk management [167]. As shown in figure 2.8, at the social and organizational levels they use a hierarchical control structure, with levels for government, regulators and associations, company, management, and staff. At all levels they map information flow. The model concentrates on operations; information from the system design and analysis process is treated as input to the operations process. At each level, they model the factors involved using event chains, with links to the event chains at the level below. Notice that they still assume there is a root cause and causal chain of events. A generalization of the Rasmussen and Svedung model, which overcomes these limitations, is presented in chapter 4.

Once again, a new assumption is needed to make progress in learning how to design and operate safer systems:

New Assumption 2: *Accidents are complex processes involving the entire sociotechnical system. Traditional event-chain models cannot describe this process adequately.*

Most of the accident models underlying safety engineering today stem from the days when the types of systems we were building and the context in which they were built were much simpler. As noted in chapter 1, new technology and social factors are making fundamental changes in the etiology of accidents, requiring changes in the explanatory mechanisms used to understand them and in the engineering techniques applied to prevent them.

Event-based models are limited in their ability to represent accidents as complex processes, particularly at representing systemic accident factors such as structural deficiencies in the organization, management deficiencies, and flaws in the safety culture of the company or industry. We need to understand how the whole system, including the organizational and social components, operating together, led to the loss. While some extensions to event-chain models have been proposed, all are unsatisfactory in important ways.

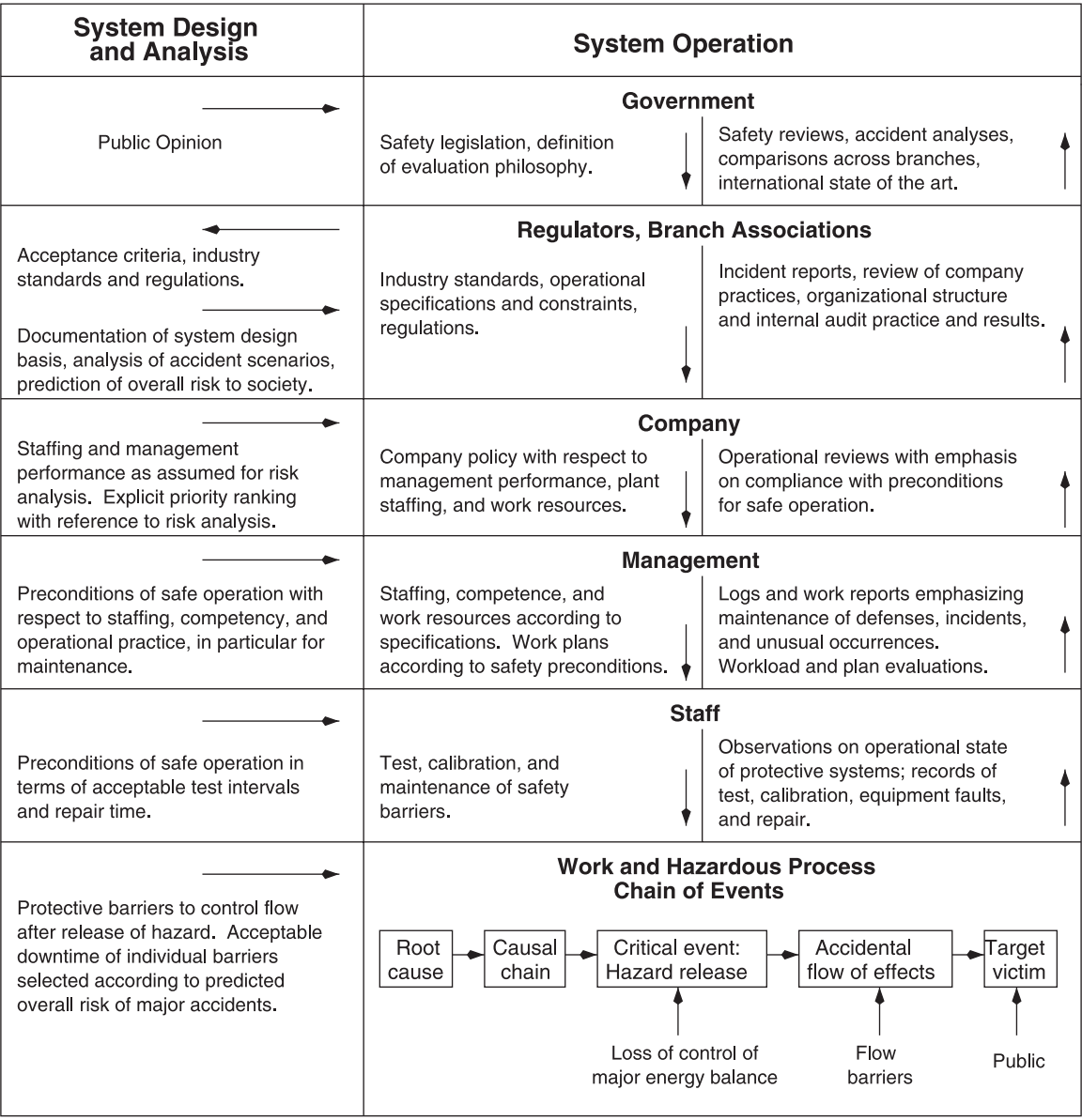


Figure 2.8
The Rasmussen/Svedung model of risk management.

An accident model should encourage a broad view of accident mechanisms that expands the investigation beyond the proximate events: A narrow focus on operator actions, physical component failures, and technology may lead to ignoring some of the most important factors in terms of preventing future accidents. The whole concept of "root cause" needs to be reconsidered.

2.3 Limitations of Probabilistic Risk Assessment

Assumption 3: *Probabilistic risk analysis based on event chains is the best way to assess and communicate safety and risk information.*

The limitations of event-chain models are reflected in the current approaches to quantitative risk assessment, most of which use trees or other forms of event chains. Probabilities (or probability density functions) are assigned to the events in the chain and an overall likelihood of a loss is calculated.

In performing a probabilistic risk assessment (PRA), initiating events in the chain are usually assumed to be mutually exclusive. While this assumption simplifies the mathematics, it may not match reality. As an example, consider the following description of an accident chain for an offshore oil platform:

> An initiating event is an event that triggers an accident sequence—e.g., a wave that exceeds the jacket's capacity that, in turn, triggers a blowout that causes failures of the foundation. As initiating events, they are mutually exclusive; only one of them starts the accident sequence. A catastrophic platform failure can start by failure of the foundation, failure of the jacket, or failure of the deck. These initiating failures are also (by definition) mutually exclusive and constitute the basic events of the [probabilistic risk assessment] model in its simplest form. [152, p. 121]

The selection of the failure of the foundation, jacket, or deck as the initiating event is arbitrary, as we have seen, and eliminates from consideration prior events leading to them such as manufacturing or construction problems. The failure of the foundation, for example, might be related to the use of inferior construction materials, which in turn might be related to budget deficiencies or lack of government oversight.

In addition, there does not seem to be any reason for assuming that initiating failures are mutually exclusive and that only one starts the accident, except perhaps again to simplify the mathematics. In accidents, seemingly independent failures may have a common systemic cause (often not a failure) that results in coincident failures. For example, the same pressures to use inferior materials in the foundation may result in their use in the jacket and the deck, leading to a wave causing coincident, dependent failures in all three. Alternatively, the design of the foundation—a systemic factor rather than a failure event—may lead to pressures on the jacket and

deck when stresses cause deformities in the foundation. Treating such events as independent may lead to unrealistic risk assessments.

In the Bhopal accident, the vent scrubber, flare tower, water spouts, refrigeration unit, and various monitoring instruments were all out of operation simultaneously. Assigning probabilities to all these seemingly unrelated events and assuming independence would lead one to believe that this accident was merely a matter of a once-in-a-lifetime coincidence. A probabilistic risk assessment based on an event chain model most likely would have treated these conditions as independent failures and then calculated their coincidence as being so remote as to be beyond consideration. Reason, in his popular Swiss Cheese Model of accident causation based on defense in depth, does the same, arguing that in general "the chances of such a trajectory of opportunity finding loopholes in all the defences at any one time is very small indeed" [172, p. 208]. As suggested earlier, a closer look at Bhopal and, indeed, most accidents paints a quite different picture and shows these were not random failure events but were related to engineering and management decisions stemming from common systemic factors.

Most accidents in well-designed systems involve two or more low-probability events occurring in the worst possible combination. When people attempt to predict system risk, they explicitly or implicitly multiply events with low probability—assuming independence—and come out with impossibly small numbers, when, in fact, the events are dependent. This dependence may be related to common systemic factors that do not appear in an event chain. Machol calls this phenomenon the *Titanic coincidence* [131].[8]

A number of "coincidences" contributed to the *Titanic* accident and the subsequent loss of life. For example, the captain was going far too fast for existing conditions, a proper watch for icebergs was not kept, the ship was not carrying enough lifeboats, lifeboat drills were not held, the lifeboats were lowered properly but arrangements for manning them were insufficient, and the radio operator on a nearby ship was asleep and so did not hear the distress call. Many of these events or conditions may be considered independent but appear less so when we consider that overconfidence due to incorrect engineering analyses about the safety and unsinkability of the ship most likely contributed to the excessive speed, the lack of a proper watch, and the insufficient number of lifeboats and drills. That the collision occurred at night contributed to the iceberg not being easily seen, made abandoning ship more difficult than it would have been during the day, and was a factor in why

8. Watt defined a related phenomenon he called the *Titanic effect* to explain the fact that major accidents are often preceded by a belief that they cannot happen. The Titanic effect says that the magnitude of disasters decreases to the extent that people believe that disasters are possible and plan to prevent them or to minimize their effects [204].

the nearby ship's operator was asleep [135]. Assuming independence here leads to a large underestimate of the true risk.

Another problem in probabilistic risk assessment (PRA) is the emphasis on failure events—design errors are usually omitted and only come into the calculation indirectly through the probability of the failure event. Accidents involving dysfunctional interactions among non-failing (operational) components—that is, component interaction accidents—are usually not considered. Systemic factors also are not reflected. In the offshore oil platform example at the beginning of this section, the true probability density function for the failure of the deck might reflect a poor design for the conditions the deck must withstand (a human design error) or, as noted earlier, the use of inadequate construction materials due to lack of government oversight or project budget limitations.

When historical data are used to determine the failure probabilities used in the PRA, non-failure factors, such as design errors or unsafe management decisions, may differ between the historic systems from which the data was derived and the system under consideration. It is possible (and obviously desirable) for each PRA to include a description of the conditions under which the probabilities were derived. If such a description is not included, it may not be possible to determine whether conditions in the platform being evaluated differ from those built previously that might significantly alter the risk. The introduction of a new design feature or of active control by a computer might greatly affect the probability of failure and the usefulness of data from previous experience then becomes highly questionable.

The most dangerous result of using PRA arises from considering only immediate physical failures. Latent design errors may be ignored and go uncorrected due to overconfidence in the risk assessment. An example, which is a common but dangerous practice judging from its implication in a surprising number of accidents, is wiring a valve to detect only that power has been applied to open or close it and not that the valve position has actually changed. In one case, an Air Force system included a relief valve to be opened by the operator to protect against overpressurization [3]. A second, backup relief valve was installed in case the primary valve failed. The operator needed to know that the first valve had not opened, however, in order to determine that the backup valve must be activated. One day, the operator issued a command to open the primary valve. The position indicator and open indicator lights both illuminated but the primary relief valve was *not* open. The operator, thinking the primary valve had opened, did not activate the backup valve and an explosion occurred.

A post-accident investigation discovered that the indicator light circuit was wired to indicate *presence of power* at the valve, but it did not indicate valve *position*. Thus, the indicator showed only that the activation button had been pushed, not that the

valve had operated. An extensive probabilistic risk assessment of this design had correctly assumed a low probability of simultaneous failure for the two relief valves, but had ignored the possibility of a design error in the electrical wiring: The probability of that design error was not quantifiable. If it had been identified, of course, the proper solution would have been to eliminate the design error, not to assign a probability to it. The same type of design flaw was a factor in the Three Mile Island accident: An indicator misleadingly showed that a discharge valve had been ordered closed but not that it had actually closed. In fact, the valve was blocked in an open position.

In addition to these limitations of PRA for electromechanical systems, current methods for quantifying risk that are based on combining probabilities of individual component failures and mutually exclusive events are not appropriate for systems controlled by software and by humans making cognitively complex decisions, and there is no effective way to incorporate management and organizational factors, such as flaws in the safety culture, despite many well-intentioned efforts to do so. As a result, these critical factors in accidents are often omitted from risk assessment because analysts do not know how to obtain a "failure" probability, or alternatively, a number is pulled out of the air for convenience. If we knew enough to measure these types of design flaws, it would be better to fix them than to try to measure them.

Another possibility for future progress is usually not considered:

New Assumption 3: *Risk and safety may be best understood and communicated in ways other than probabilistic risk analysis.*

Understanding risk is important in decision making. Many people assume that risk information is most appropriately communicated in the form of a probability. Much has been written, however, about the difficulty people have in interpreting probabilities [97]. Even if people could use such values appropriately, the tools commonly used to compute these quantities, which are based on computing probabilities of failure events, have serious limitations. An accident model that is not based on failure events, such as the one introduced in this book, could provide an entirely new basis for understanding and evaluating safety and, more generally, risk.

2.4 The Role of Operators in Accidents

Assumption 4: *Most accidents are caused by operator error. Rewarding safe behavior and punishing unsafe behavior will eliminate or reduce accidents significantly.*

As we have seen, the definition of "caused by" is debatable. But the fact remains that if there are operators in the system, they are most likely to be blamed for an

accident. This phenomenon is not new. In the nineteenth century, coupling accidents on railroads were one of the main causes of injury and death to railroad workers [79]. In the seven years between 1888 and 1894, 16,000 railroad workers were killed in coupling accidents and 170,000 were crippled. Managers claimed that such accidents were due only to worker error and negligence, and therefore nothing could be done aside from telling workers to be more careful. The government finally stepped in and required that automatic couplers be installed. As a result, fatalities dropped sharply. According to the June 1896 (three years after Congress acted on the problem) issue of *Scientific American*:

> Few battles in history show so ghastly a fatality. A large percentage of these deaths were caused by the use of imperfect equipment by the railroad companies; twenty years ago it was practically demonstrated that cars could be automatically coupled, and that it was no longer necessary for a railroad employee to imperil his life by stepping between two cars about to be connected. In response to appeals from all over, the U.S. Congress passed the Safety Appliance Act in March 1893. It has or will cost the railroads $50,000,000 to fully comply with the provisions of the law. Such progress has already been made that the death rate has dropped by 35 per cent.

2.4.1 Do Operators Cause Most Accidents?

The tendency to blame the operator is not simply a nineteenth century problem, but persists today. During and after World War II, the Air Force had serious problems with aircraft accidents: From 1952 to 1966, for example, 7,715 aircraft were lost and 8,547 people killed [79]. Most of these accidents were blamed on pilots. Some aerospace engineers in the 1950s did not believe the cause was so simple and argued that safety must be designed and built into aircraft just as are performance, stability, and structural integrity. Although a few seminars were conducted and papers written about this approach, the Air Force did not take it seriously until they began to develop intercontinental ballistic missiles: there were no pilots to blame for the frequent and devastating explosions of these liquid-propellant missiles. In having to confront factors other than pilot error, the Air Force began to treat safety as a system problem, and System Safety programs were developed to deal with them. Similar adjustments in attitude and practice may be forced in the future by the increasing use of unmanned autonomous aircraft and other automated systems.

It is still common to see statements that 70 percent to 80 percent of aircraft accidents are caused by pilot error or that 85 percent of work accidents are due to unsafe acts by workers rather than unsafe conditions. However, closer examination shows that the data may be biased and incomplete: the less that is known about an accident, the most likely it will be attributed to operator error [93]. Thorough investigation of serious accidents almost invariably finds other factors.

Part of the problem stems from the use of the chain-of-events model in accident investigation because it is difficult to find an *event* preceding and causal to the operator behavior, as mentioned earlier. If the problem is in the system design, there is no proximal event to explain the error, only a flawed decision during system design.

Even if a technical failure precedes the human action, the tendency is to put the blame on an inadequate response to the failure by an operator. Perrow claims that even in the best of industries, there is rampant attribution of accidents to operator error, to the neglect of errors by designers or managers [155]. He cites a U.S. Air Force study of aviation accidents demonstrating that the designation of human error (pilot error in this case) is a convenient classification for mishaps whose real cause is uncertain, complex, or embarrassing to the organization.

Beside the fact that operator actions represent a convenient stopping point in an event chain, other reasons for the operator error statistics include: (1) operator actions are generally reported only when they have a negative impact on safety and not when they are responsible for preventing accidents; (2) blame may be based on unrealistic expectations that operators can overcome every emergency; (3) operators may have to intervene at the limits of system behavior when the consequences of not succeeding are likely to be serious and often involve a situation the designer never anticipated and was not covered by the operator's training; and (4) hindsight often allows us to identify a better decision in retrospect, but detecting and correcting potential errors before they have been made obvious by an accident is far more difficult.[9]

2.4.2 Hindsight Bias

The psychological phenomenon called *hindsight bias* plays such an important role in attribution of causes to accidents that it is worth spending time on it. The report on the Clapham Junction railway accident in Britain concluded:

> There is almost no human action or decision that cannot be made to look flawed and less sensible in the misleading light of hindsight. It is essential that the critic should keep himself constantly aware of that fact. [82, pg. 147]

After an accident, it is easy to see where people went wrong, what they should have done or not done, to judge people for missing a piece of information that turned out to be critical, and to see exactly the kind of harm that they should have foreseen or prevented [51]. Before the event, such insight is difficult and, perhaps, impossible.

9. The attribution of operator error as the cause of accidents is discussed more thoroughly in *Safeware* (chapter 5).

Dekker [51] points out that hindsight allows us to:

- Oversimplify causality because we can start from the outcome and reason backward to presumed or plausible "causes."
- Overestimate the likelihood of the outcome—and people's ability to foresee it—because we already know what the outcome is.
- Overrate the role of rule or procedure "violations." There is always a gap between written guidance and actual practice, but this gap almost never leads to trouble. It only takes on causal significance once we have a bad outcome to look at and reason about.
- Misjudge the prominence or relevance of data presented to people at the time.
- Match outcome with the actions that went before it. If the outcome was bad, then the actions leading up to it must have also been bad—missed opportunities, bad assessments, wrong decisions, and misperceptions.

Avoiding hindsight bias requires changing our emphasis in analyzing the role of humans in accidents from what they did wrong to why it made sense for them to act the way they did.

2.4.3 The Impact of System Design on Human Error

All human activity takes place within and is influenced by the environment, both physical and social, in which it takes place. It is, therefore, often very difficult to separate system design error from operator error: In highly automated systems, the operator is often at the mercy of the system design and operational procedures. One of the major mistakes made by the operators at Three Mile Island was following the procedures provided to them by the utility. The instrumentation design also did not provide the information they needed to act effectively in recovering from the hazardous state [99].

In the lawsuits following the 1995 B757 Cali accident, American Airlines was held liable for the crash based on the Colombian investigators blaming crew error entirely for the accident. The official accident investigation report cited the following four causes for the loss [2]:

1. The flightcrew's failure to adequately plan and execute the approach to runway 19 and their inadequate use of automation.
2. Failure of the flightcrew to discontinue their approach, despite numerous cues alerting them of the inadvisability of continuing the approach.
3. The lack of situational awareness of the flightcrew regarding vertical navigation, proximity to terrain, and the relative location of critical radio aids.

4. Failure of the flightcrew to revert to basic radio navigation at a time when the FMS-assisted navigation became confusing and demanded an excessive workload in a critical phase of the flight.

Look in particular the fourth identified cause: the blame is placed on the pilots when the automation became confusing and demanded an excessive workload rather than on the design of the automation. To be fair, the report also identifies two "contributory factors"—but *not* causes—as:

- FMS logic that dropped all intermediate fixes from the display(s) in the event of execution of a direct routing.
- FMS-generated navigational information that used a different naming convention from that published in navigational charts.

These two "contributory factors" are highly related to the third cause—the pilots' "lack of situational awareness." Even using an event-chain model of accidents, the FMS-related events preceded and contributed to the pilot errors. There seems to be no reason why, at the least, they should be treated any different than the labeled "causes." There were also many other factors in this accident that were not reflected in either the identified causes or contributory factors.

In this case, the Cali accident report conclusions were challenged in court. A U.S. appeals court rejected the conclusion of the report about the four causes of the accident [13], which led to a lawsuit by American Airlines in a federal court in which American alleged that components of the automated aircraft system made by Honeywell Air Transport Systems and Jeppesen Sanderson helped cause the crash. American blamed the software, saying Jeppesen stored the location of the Cali airport beacon in a different file from most other beacons. Lawyers for the computer companies argued that the beacon code could have been properly accessed and that the pilots were in error. The jury concluded that the two companies produced a defective product and that Jeppesen was 17 percent responsible, Honeywell was 8 percent at fault, and American was held to be 75 percent responsible [7]. While such distribution of responsibility may be important in determining how much each company will have to pay, it is arbitrary and does not provide any important information with respect to accident prevention in the future. The verdict is interesting, however, because the jury rejected the oversimplified notion of causality being argued. It was also one of the first cases not settled out of court where the role of software in the loss was acknowledged.

This case, however, does not seem to have had much impact on the attribution of pilot error in later aircraft accidents.

Part of the problem is engineers' tendency to equate people with machines. Human "failure" usually is treated the same as a physical component failure—a

deviation from the performance of a specified or prescribed sequence of actions. This definition is equivalent to that of machine failure. Alas, human behavior is much more complex than machines.

As many human factors experts have found, instructions and written procedures are almost never followed exactly as operators try to become more efficient and productive and to deal with time pressures [167]. In studies of operators, even in such highly constrained and high-risk environments as nuclear power plants, modification of instructions is repeatedly found [71, 201, 213]. When examined, these violations of rules appear to be quite rational, given the workload and timing constraints under which the operators must do their job. The explanation lies in the basic conflict between error viewed as a deviation from *normative procedure* and error viewed as a deviation from the rational and normally used *effective procedure* [169].

One implication is that following an accident, it will be easy to find someone involved in the dynamic flow of events that has violated a formal rule by following *established practice* rather than *specified practice.* Given the frequent deviation of established practice from normative work instructions and rules, it is not surprising that operator "error" is found to be the cause of 70 percent to 80 percent of accidents. As noted in the discussion of assumption 2, a root cause is often selected because that event involves a deviation from a standard.

2.4.4 The Role of Mental Models

The updating of human mental models plays a significant role here (figure 2.9). Both the designer and the operator will have their own mental models of the plant. It is quite natural for the designer's and operator's models to differ and even for both to have significant differences from the actual plant as it exists. During development, the designer evolves a model of the plant to the point where it can be built. The *designer's model* is an idealization formed *before* the plant is constructed. Significant differences may exist between this ideal model and the actual constructed system. Besides construction variances, the designer always deals with ideals or averages, not with the actual components themselves. Thus, a designer may have a model of a valve with an average closure time, while real valves have closure times that fall somewhere along a continuum of timing behavior that reflects manufacturing and material differences. The designer's idealized model is used to develop operator work instructions and training. But the actual system may differ from the designer's model because of manufacturing and construction variances and evolution and changes over time.

The *operator's model* of the system will be based partly on formal training created from the designer's model and partly on experience with the system. The operator must cope with the system as it is constructed and not as it may have been

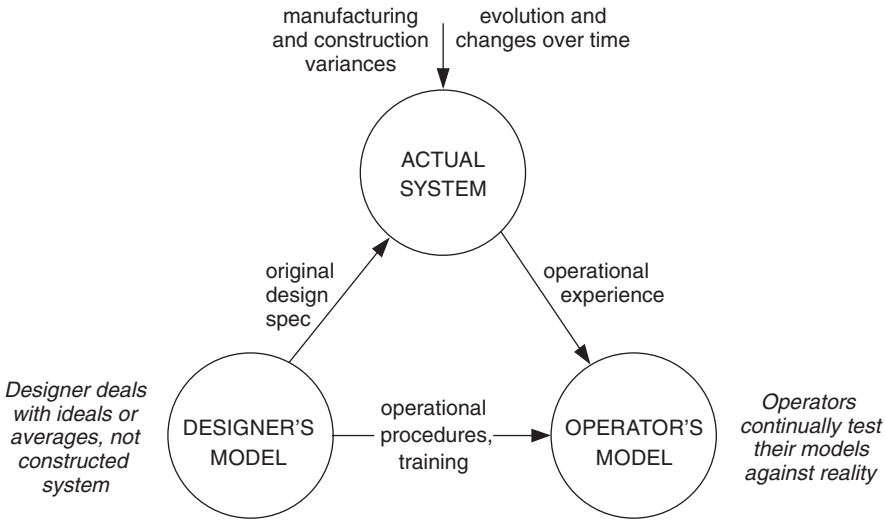


Figure 2.9
The relationship between mental models.

envisioned. As the physical system changes and evolves over time, the operator's model and operational procedures must change accordingly. While the formal procedures, work instructions, and training will be updated periodically to reflect the current operating environment, there is necessarily always a time lag. In addition, the operator may be working under time and productivity pressures that are not reflected in the idealized procedures and training.

Operators use feedback to update their mental models of the system as the system evolves. The only way for the operator to determine that the system has changed and that his or her mental model must be updated is through experimentation: To learn where the boundaries of safe behavior currently are, occasionally they must be crossed.

Experimentation is important at all levels of control [166]. For manual tasks where the optimization criteria are speed and smoothness, the limits of acceptable adaptation and optimization can only be known from the error experienced when occasionally crossing a limit. Errors are an integral part of maintaining a skill at an optimal level and a necessary part of the feedback loop to achieve this goal. The role of such experimentation in accidents cannot be understood by treating human errors as events in a causal chain separate from the feedback loops in which they operate.

At higher levels of cognitive control and supervisory decision making, experimentation is needed for operators to update procedures to handle changing

conditions or to evaluate hypotheses while engaged in reasoning about the best response to unexpected situations. Actions that are quite rational and important during the search for information and test of hypotheses may appear to be unacceptable mistakes in hindsight, without access to the many details of a "turbulent" situation [169].

The ability to adapt mental models through experience in interacting with the operating system is what makes the human operator so valuable. For the reasons discussed, the operators' actual behavior may differ from the prescribed procedures because it is based on current inputs and feedback. When the deviation is correct (the designers' models are less accurate than the operators' models at that particular instant in time), then the operators are considered to be doing their job. When the operators' models are incorrect, they are often blamed for any unfortunate results, even though their incorrect mental models may have been reasonable given the information they had at the time.

Providing feedback and allowing for experimentation in system design, then, is critical in allowing operators to optimize their control ability. In the less automated system designs of the past, operators naturally had this ability to experiment and update their mental models of the current system state. Designers of highly automated systems sometimes do not understand this requirement and design automation that takes operators "out of the loop." Everyone is then surprised when the operator makes a mistake based on an incorrect mental model. Unfortunately, the reaction to such a mistake is to add even more automation and to marginalize the operators even more, thus exacerbating the problem [50].

Flawed decisions may also result from limitations in the boundaries of the operator's or designer's model. Decision makers may simply have too narrow a view of the system their decisions will impact. Recall figure 2.2 and the discussion of the *Herald of Free Enterprise* accident. The boundaries of the system model relevant to a particular decision maker may depend on the activities of several other decision makers found within the total system [167]. Accidents may result from the interaction and side effects of their decisions based on their limited model. Before an accident, it will be difficult for the individual decision makers to see the full picture during their daily operational decision making and to judge the current state of the multiple defenses and safety margins that are partly dependent on decisions made by other people in other departments and organizations [167].

Rasmussen stresses that most decisions are sound using local judgment criteria and given the time and budget pressures and short-term incentives that shape behavior. Experts do their best to meet local conditions and in the busy daily flow of activities may be unaware of the potentially dangerous side effects of their behavior. Each individual decision may appear safe and rational within the context of the individual work environments and local pressures, but may be unsafe when

considered as a whole: It is difficult—if not impossible—for any individual to judge the safety of their decisions when it is dependent on the decisions made by other people in other departments and organizations.

Decentralized decision making is, of course, required in some time-critical situations. But like all safety-critical decision making, the decentralized decisions must be made in the context of system-level information and from a total systems perspective in order to be effective in reducing accidents. One way to make distributed decision making safe is to decouple the system components in the overall system design, if possible, so that decisions do not have systemwide repercussions. Another common way to deal with the problem is to specify and train standard emergency responses. Operators may be told to sound the evacuation alarm any time an indicator reaches a certain level. In this way, safe procedures are determined at the system level and operators are socialized and trained to provide uniform and appropriate responses to crisis situations.

There are situations, of course, when unexpected conditions occur and avoiding losses requires the operators to violate the specified (and in such cases unsafe) procedures. If the operators are expected to make decisions in real time and not just follow a predetermined procedure, then they usually must have the relevant *system*-level information about the situation in order to make safe decisions. This is not required, of course, if the system design decouples the components and thus allows operators to make independent safe decisions. Such decoupling must be designed into the system, however.

Some high reliability organization (HRO) theorists have argued just the opposite. They have asserted that HROs are safe because they allow professionals at the front lines to use their knowledge and judgment to maintain safety. During crises, they argue, decision making in HROs migrates to the frontline workers who have the necessary judgment to make decisions [206]. The problem is that the assumption that frontline workers will have the necessary knowledge and judgment to make decisions is not necessarily true. One example is the friendly fire accident analyzed in chapter 5 where the pilots ignored the rules of engagement they were told to follow and decided to make real-time decisions on their own based on the inadequate information they had.

Many of the HRO theories were derived from studying safety-critical systems, such as aircraft carrier flight operations. La Porte and Consolini [107], for example, argue that while the operation of aircraft carriers is subject to the Navy's chain of command, even the lowest-level seaman can abort landings. Clearly, this local authority is necessary in the case of aborted landings because decisions must be made too quickly to go up a chain of command. But note that such low-level personnel can only make decisions in one direction, that is, they may only abort landings. In essence, they are allowed to change to an inherently safe state (a go-around)

with respect to the hazard involved. System-level information is not needed because a safe state exists that has no conflicts with other hazards, and the actions governed by these decisions and the conditions for making them are relatively simple. Aircraft carriers are usually operating in areas containing little traffic—they are decoupled from the larger system—and therefore localized decisions to abort are almost always safe and can be allowed from a larger system safety viewpoint.

Consider a slightly different situation, however, where a pilot makes a decision to go-around (abort a landing) at a busy urban airport. While executing a go-around when a clear danger exists if the pilot lands is obviously the right decision, there have been near misses when a pilot executed a go-around and came too close to another aircraft that was taking off on a perpendicular runway. The solution to this problem is not at the decentralized level—the individual pilot lacks the system-level information to avoid hazardous system states in this case. Instead, the solution must be at the system level, where the danger must be reduced by instituting different landing and takeoff procedures, building new runways, redistributing air traffic, or by making other system-level changes. We want pilots to be able to execute a go-around if they feel it is necessary, but unless the encompassing system is designed to prevent collisions, the action decreases one hazard while increasing a different one. Safety is a system property.

2.4.5 An Alternative View of Human Error

Traditional decision-making research views decisions as discrete processes that can be separated from the context in which the decisions are made and studied as an isolated phenomenon. This view is starting to be challenged. Instead of thinking of operations as predefined sequences of actions, human interaction with a system is increasingly being considered to be a continuous control task in which separate "decisions" or errors are difficult to identify.

Edwards, back in 1962, was one of the first to argue that decisions can only be understood as part of an ongoing process [63]. The state of the system is perceived in terms of possible actions, one of these actions is chosen, and the resulting response from the controlled system acts as a background for the next actions. Errors then are difficult to localize in the stream of behavior; the effects of less successful actions are a natural part of the search by the operator for optimal performance. As an example, consider steering a boat. The helmsman of ship A may see an obstacle ahead (perhaps another ship) and decide to steer the boat to the left to avoid it. The wind, current, and wave action may require the helmsman to make continual adjustments in order to hold the desired course. At some point, the other ship may also change course, making the helmsman's first decision about what would be a safe course no longer correct and needing to be revised. Steering then can be perceived as a continuous control activity or process with what is the correct and safe

behavior changing over time and with respect to the results of prior behavior. The helmsman's mental model of the effects of the actions of the sea and the assumed behavior of the other ship has to be continually adjusted.

Not only are individual unsafe actions difficult to identify in this nontraditional control model of human decision making, but the study of decision making cannot be separated from a simultaneous study of the social context, the value system in which it takes place, and the dynamic work process it is intended to control [166]. This view is the foundation of some modern trends in decision-making research, such as *dynamic decision making* [25], the new field of *naturalistic decision making* [217, 102], and the approach to safety described in this book.

As argued by Rasmussen and others, devising more effective accident models that go beyond the simple event chain and human failure models requires shifting the emphasis in explaining the role of humans in accidents from error (that is, deviations from normative procedures) to focus instead on the mechanisms and factors that shape human behavior, that is, the performance-shaping context in which human actions take place and decisions are made. Modeling human behavior by decomposing it into decisions and actions and studying it as a phenomenon isolated from the context in which the behavior takes place is not an effective way to understand behavior [167].

The alternative view requires a new approach to representing and understanding human behavior, focused not on human error and violation of rules but on the mechanisms generating behavior in the actual, dynamic context. Such as approach must take into account the work system constraints, the boundaries of acceptable performance, the need for experimentation, and the subjective criteria guiding adaptation to change. In this approach, traditional task analysis is replaced or augmented with *cognitive work analysis* [169, 202] or *cognitive task analysis* [75]. Behavior is modeled in terms of the objectives of the decision maker, the boundaries of acceptable performance, the behavior-shaping constraints of the environment (including the value system and safety constraints), and the adaptive mechanisms of the human actors.

Such an approach leads to new ways of dealing with the human contribution to accidents and human "error." Instead of trying to control human behavior by fighting deviations from specified procedures, focus should be on controlling behavior by identifying the boundaries of safe performance (the behavioral safety constraints), by making the boundaries explicit and known, by giving opportunities to develop coping skills at the boundaries, by designing systems to support safe optimization and adaptation of performance in response to contextual influences and pressures, by providing means for identifying potentially dangerous side effects of individual decisions in the network of decisions over the entire system, by

designing for error tolerance (making errors observable and reversible before safety constraints are violated) [167], and by counteracting the pressures that drive operators and decision makers to violate safety constraints.

Once again, future progress in accident reduction requires tossing out the old assumption and substituting a new one:

New Assumption 4: *Operator behavior is a product of the environment in which it occurs. To reduce operator "error" we must change the environment in which the operator works.*

Human behavior is always influenced by the environment in which it takes place. Changing that environment will be much more effective in changing operator error than the usual behaviorist approach of using reward and punishment. Without changing the environment, human error cannot be reduced for long. We design systems in which operator error is inevitable, and then blame the operator and not the system design.

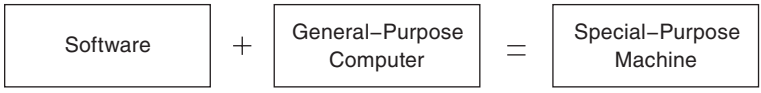
As argued by Rasmussen and others, devising more effective accident causality models requires shifting the emphasis in explaining the role that humans play in accidents from error (deviations from normative procedures) to focus on the mechanisms and factors that shape human behavior, that is the performance-shaping features and context in which human actions take place and decisions are made. Modeling behavior by decomposing it into decisions and actions or events, which most all current accident models do, and studying it as a phenomenon isolated from the context in which the behavior takes place is not an effective way to understand behavior [167].

2.5 The Role of Software in Accidents

Assumption 5: *Highly reliable software is safe.*

The most common approach to ensuring safety when the system includes software is to try to make the software highly reliable. To help readers who are not software professionals see the flaws in this assumption, a few words about software in general may be helpful.

The uniqueness and power of the digital computer over other machines stems from the fact that, for the first time, we have a general-purpose machine:

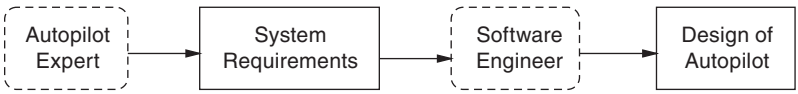


We no longer need to build a mechanical or analog autopilot from scratch, for example, but simply to write down the "design" of an autopilot in the form of instructions or steps to accomplish the desired goals. These steps are then loaded into the computer, which, while executing the instructions, in effect *becomes* the special-purpose machine (the autopilot). If changes are needed, the instructions can be changed and the same physical machine (the computer hardware) is used instead of having to build a different physical machine from scratch. Software in essence is the *design of a machine abstracted from its physical realization.* In other words, the logical design of a machine (the software) is separated from the physical design of that machine (the computer hardware).

Machines that previously were physically impossible or impractical to build become feasible, and the design of a machine can be changed quickly without going through an entire retooling and manufacturing process. In essence, the manufacturing phase is eliminated from the lifecycle of these machines: the physical parts of the machine (the computer hardware) can be reused, leaving only the design and verification phases. The design phase also has changed: The designer can concentrate on identifying the steps to be achieved without having to worry about how those steps will be realized physically.

These advantages of using computers (along with others specific to particular applications, such as reduced size and weight) have led to an explosive increase in their use, including their introduction into potentially dangerous systems. There are, however, some potential disadvantages of using computers and some important changes that their use introduces into the traditional engineering process that are leading to new types of accidents as well as creating difficulties in investigating accidents and preventing them.

One of the most important changes is that with computers, the design of the special purpose machine is usually created by someone who is not an expert on designing such machines. The autopilot design expert, for example, decides how the autopilot should work, and then provides that information to a software engineer, who is an expert in software design but not autopilots. It is the software engineer who then creates the detailed design of the autopilot. The extra communication step between the engineer and the software developer is the source of the most serious problems with software today.



It should not be surprising, then, that most errors found in operational software can be traced to requirements flaws, particularly incompleteness. Completeness is a

quality often associated with requirements but rarely defined. The most appropriate definition in the context of this book has been proposed by Jaffe: Software requirements specifications are complete if they are sufficient to distinguish the desired behavior of the software from that of any other undesired program that might be designed [91].

Nearly all the serious accidents in which software has been involved in the past twenty years can be traced to requirements flaws, not coding errors. The requirements may reflect incomplete or wrong assumptions

- About the operation of the system components being controlled by the software (for example, how quickly the component can react to a software-generated control command) or
- About the required operation of the computer itself

In the Mars Polar Lander loss, the software requirements did not include information about the potential for the landing leg sensors to generate noise or, alternatively, to ignore any inputs from the sensors while the spacecraft was more than forty meters above the planet surface. In the batch chemical reactor accident, the software engineers were never told to open the water valve before the catalyst valve and apparently thought the ordering was therefore irrelevant.

The problems may also stem from unhandled controlled-system states and environmental conditions. An F-18 was lost when a mechanical failure in the aircraft led to the inputs arriving faster than expected, which overwhelmed the software [70]. Another F-18 loss resulted from the aircraft getting into an attitude that the engineers had assumed was impossible and that the software was not programmed to handle.

In these cases, simply trying to get the software "correct" in terms of accurately implementing the requirements will not make it safer. Software may be highly reliable and correct and still be unsafe when:

- The software correctly implements the requirements, but the specified behavior is unsafe from a system perspective.
- The software requirements do not specify some particular behavior required for system safety (that is, they are incomplete).
- The software has unintended (and unsafe) behavior beyond what is specified in the requirements.

If the problems stem from the software doing what the software engineer thought it should do when that is not what the original design engineer wanted, the use of integrated product teams and other project management schemes to help with communication are useful. The most serious problems arise, however, when *nobody*

understands what the software should do or even what it should not do. We need better techniques to assist in determining these requirements.

There is not only anecdotal but some hard data to support the hypothesis that safety problems in software stem from requirements flaws and not coding errors. Lutz examined 387 software errors uncovered during integration and system testing of the Voyager and Galileo spacecraft [130]. She concluded that the software errors identified as potentially hazardous to the system tended to be produced by different error mechanisms than non-safety-related software errors. She showed that for these two spacecraft, the safety-related software errors arose most commonly from (1) discrepancies between the documented requirements specifications and the requirements needed for correct functioning of the system and (2) misunderstandings about the software's interface with the rest of the system. They did not involve coding errors in implementing the documented requirements.

Many software requirements problems arise from what could be called the *curse of flexibility*. The computer is so powerful and so useful because it has eliminated many of the physical constraints of previous machines. This is both its blessing and its curse: We no longer have to worry about the physical realization of our designs, but we also no longer have physical laws that limit the complexity of our designs. Physical constraints enforce discipline on the design, construction, and modification of our design artifacts. Physical constraints also control the complexity of what we build. With software, the limits of what is *possible* to accomplish are different than the limits of what can be accomplished *successfully* and *safely*—the limiting factors change from the structural integrity and physical constraints of our materials to limits on our intellectual capabilities.

It is possible and even quite easy to build software that we cannot understand in terms of being able to determine how it will behave under all conditions. We can construct software (and often do) that goes beyond human intellectual limits. The result has been an increase in component interaction accidents stemming from intellectual unmanageability that allows potentially unsafe interactions to go undetected during development. The software often controls the interactions among the system components so its close relationship with component interaction accidents should not be surprising. But this fact has important implications for how software must be engineered when it controls potentially unsafe systems or products: Software or system engineering techniques that simply ensure software reliability or correctness (consistency of the code with the requirements) will have little or no impact on safety.

Techniques that *are* effective will rest on a new assumption:

New Assumption 5: *Highly reliable software is not necessarily safe. Increasing software reliability or reducing implementation errors will have little impact on safety.*

2.6 Static versus Dynamic Views of Systems

Assumption 6: *Major accidents occur from the chance simultaneous occurrence of random events.*

Most current safety engineering techniques suffer from the limitation of considering only the events underlying an accident and not the entire accident *process.* Accidents are often viewed as some unfortunate coincidence of factors that come together at one particular point in time and lead to the loss. This belief arises from too narrow a view of the causal time line. Looking only at the immediate time of the Bhopal MIC release, it does seem to be a coincidence that the refrigeration system, flare tower, vent scrubber, alarms, water curtain, and so on had all been inoperable at the same time. But viewing the accident through a larger lens makes it clear that the causal factors were all related to systemic causes that had existed for a long time.

Systems are not static. Rather than accidents being a chance occurrence of multiple independent events, they tend to involve a migration to a state of increasing risk over time [167]. A point is reached where an accident is inevitable unless the high risk is detected and reduced. The particular events involved at the time of the loss are somewhat irrelevant: if those events had not occurred, something else would have led to the loss. This concept is reflected in the common observation that a loss was "an accident waiting to happen." The proximate cause of the *Columbia* Space Shuttle loss was the foam coming loose from the external tank and damaging the reentry heat control structure. But many potential problems that could have caused the loss of the Shuttle had preceded this event and an accident was avoided by luck or unusual circumstances. The economic and political pressures led the Shuttle program to migrate to a state where any slight deviation could have led to a loss [117].

Any approach to enhancing safety that includes the social system and humans must account for adaptation. To paraphrase a familiar saying, the only constant is that nothing ever remains constant. Systems and organizations continually experience change as adaptations are made in response to local pressures and short-term productivity and cost goals. People adapt to their environment or they change their environment to better suit their purposes. A corollary to this propensity for systems and people to adapt over time is that safety defenses are likely to degenerate systematically through time, particularly when pressure toward cost-effectiveness and increased productivity is the dominant element in decision making. Rasmussen noted that the critical factor here is that such adaptation is not a random process—it is an optimization process depending on search strategies—and thus should be predictable and potentially controllable [167].

Woods has stressed the importance of adaptation in accidents. He describes organizational and human failures as breakdowns in adaptations directed at coping with complexity, and accidents as involving a "drift toward failure as planned defenses erode in the face of production pressures and change" [214].

Similarly, Rasmussen has argued that major accidents are often caused not by a coincidence of independent failures but instead reflect a systematic migration of organizational behavior to the boundaries of safe behavior under pressure toward cost-effectiveness in an aggressive, competitive environment [167]. One implication of this viewpoint is that the struggle for a good safety culture will never end because it must continually fight against the functional pressures of the work environment. Improvement of the safety culture will therefore require an analytical approach directed toward the behavior-shaping factors in the environment. A way of achieving this goal is described in part III.

Humans and organizations can adapt and still maintain safety as long as they stay within the area bounded by safety constraints. But in the search for optimal operations, humans and organizations will close in on and explore the boundaries of established practice. Such exploration implies the risk of occasionally crossing the limits of safe practice unless the constraints on safe behavior are enforced.

The natural migration toward the boundaries of safe behavior, according to Rasmussen, is complicated by the fact that it results from the decisions of multiple people, in different work environments and contexts within the overall sociotechnical system, all subject to competitive or budgetary stresses and each trying to optimize their decisions within their own immediate context. Several decision makers at different times, in different parts of the company or organization, all striving locally to optimize cost effectiveness may be preparing the stage for an accident, as illustrated by the Zeebrugge ferry accident (see figure 2.2) and the friendly fire accident described in chapter 5. The dynamic flow of events can then be released by a single act.

Our new assumption is therefore:

New Assumption 6: *Systems will tend to migrate toward states of higher risk. Such migration is predictable and can be prevented by appropriate system design or detected during operations using leading indicators of increasing risk.*

To handle system adaptation over time, our causal models and safety techniques must consider the *processes* involved in accidents and not simply events and conditions: Processes control a sequence of events and describe system and human behavior as it changes and adapts over time rather than considering individual events and human actions. To talk about the cause or causes of an accident makes no sense in this systems or process view of accidents. As Rasmussen argues, deterministic causal models are inadequate to explain the organizational and social

factors in highly adaptive sociotechnical systems. Instead, accident causation must be viewed as a complex process involving the entire sociotechnical system including legislators, government agencies, industry associations and insurance companies, company management, technical and engineering personnel, operations, and so on [167].

2.7 The Focus on Determining Blame

Assumption 7: *Assigning blame is necessary to learn from and prevent accidents or incidents.*

Beyond the tendency to blame operators described under assumption 3, other types of subjectivity in ascribing cause exist. Rarely are all the causes of an accident perceived identically by everyone involved, including engineers, managers, operators, union officials, insurers, lawyers, politicians, the press, the state, and the victims and their families. Such conflicts are typical in situations that involve normative, ethical, and political considerations about which people may legitimately disagree. Some conditions may be considered unnecessarily hazardous by one group yet adequately safe and necessary by another. In addition, judgments about the cause of an accident may be affected by the threat of litigation or by conflicting interests.

Research data validates this hypothesis. Various studies have found the selection of a cause(s) depends on characteristics of the victim and of the analyst (e.g., hierarchical status, degree of involvement, and job satisfaction) as well as on the relationships between the victim and the analyst and on the severity of the accident [112].

For example, one study found that workers who were satisfied with their jobs and who were integrated into and participating in the enterprise attributed accidents mainly to personal causes. In contrast, workers who were not satisfied and who had a low degree of integration and participation more often cited nonpersonal causes that implied that the enterprise was responsible [112]. Another study found differences in the attribution of accident causes among victims, safety managers, and general managers. Other researchers have suggested that accidents are attributed to factors in which the individuals are less directly involved. A further consideration may be position in the organization: The lower the position in the hierarchy, the greater the tendency to blame accidents on factors linked to the organization; individuals who have a high position in the hierarchy tend to blame workers for accidents [112].

There even seem to be differences in causal attribution between accidents and incidents: Accident investigation data on near-miss (incident) reporting suggest that causes for these events are mainly attributed to technical deviations while similar events that result in losses are more often blamed on operator error [62, 100].

Causal identification may also be influenced by the data collection methods. Data are usually collected in the form of textual descriptions of the sequence of events of the accident, which, as we have seen, tend to concentrate on obvious conditions or events closely preceding the accident in time and tend to leave out less obvious or indirect events and factors. There is no simple solution to this inherent bias: On one hand, report forms that do not specifically ask for nonproximal factors often do not elicit them while, on the other hand, more directive report forms that do request particular information may limit the categories or conditions considered [101].

Other factors affecting causal filtering in accident and incident reports may be related to the design of the reporting system itself. For example, the NASA Aviation Safety Reporting System (ASRS) has a category that includes nonadherence to FARs (Federal Aviation Regulations). In a NASA study of reported helicopter incidents and accidents over a nine-year period, this category was by far the largest category cited [81]. The NASA study concluded that the predominance of FAR violations in the incident data may reflect the motivation of the ASRS reporters to obtain immunity from perceived or real violations of FARs and not necessarily the true percentages.

A final complication is that human actions always involve some interpretation of the person's goals and motives. The individuals involved may be unaware of their actual goals and motivation or may be subject to various types of pressures to reinterpret their actions. Explanations by accident analysts after the fact may be influenced by their own mental models or additional goals and pressures.

Note the difference between an explanation based on goals and one based on motives: a goal represents an end state while a motive explains *why* that end state was chosen. Consider the hypothetical case where a car is driven too fast during a snowstorm and slides into a telephone pole. An explanation based on goals for this chain of events might include the fact that the driver wanted to get home quickly. An explanation based on motives might include the fact that guests were coming for dinner and the driver had to prepare the food before they arrived.

Explanations based on goals and motives depend on assumptions that cannot be directly measured or observed by the accident investigator. Leplat illustrates this dilemma by describing three different motives for the event *"operator sweeps the floor"*: (1) the floor is dirty, (2) the supervisor is present, or (3) the machine is broken and the operator needs to find other work [113]. Even if the people involved survive the accident, true goals and motives may not be revealed for a variety of reasons.

Where does all this leave us? There are two possible reasons for conducting an accident investigation: (1) to assign blame for the accident and (2) to understand why it happened so that future accidents can be prevented. When the goal is to assign blame, the backward chain of events considered often stops when someone or something appropriate to blame is found, such as the baggage handler in the

DC-10 case or the maintenance worker at Bhopal. As a result, the selected initiating event may provide too superficial an explanation of why the accident occurred to prevent similar losses in the future.

As another example, stopping at the O-ring failure in the *Challenger* accident and fixing that particular design flaw would not have eliminated the systemic flaws that could lead to accidents in the future. For *Challenger*, examples of those systemic problems include flawed decision making and the political and economic pressures that led to it, poor problem reporting, lack of trend analysis, a "silent" or ineffective safety program, communication problems, etc. None of these are "events" (although they may be manifested in particular events) and thus do not appear in the chain of events leading to the accident. Wisely, the authors of the *Challenger* accident report used an event chain only to identify the proximate physical cause and not the reasons those events occurred, and the report's recommendations led to many important changes at NASA or at least attempts to make such changes.

Twenty years later, another Space Shuttle was lost. While the proximate cause for the *Columbia* accident (foam hitting the wing of the orbiter) was very different than that for *Challenger*, many of the systemic causal factors were similar and reflected either inadequate fixes of these factors after the *Challenger* accident or their reemergence in the years between these losses [117].

Blame is not an engineering concept; it is a legal or moral one. Usually there is no objective criterion for distinguishing one factor or several factors from other factors that contribute to an accident. While lawyers and insurers recognize that many factors contribute to a loss event, for practical reasons and particularly for establishing liability, they often oversimplify the causes of accidents and identify what they call the *proximate* (immediate or direct) cause. The goal is to determine the parties in a dispute that have the legal liability to pay damages, which may be affected by the ability to pay or by public policy considerations, such as discouraging company management or even an entire industry from acting in a particular way in the future.

When learning how to engineer safer systems is the goal rather than identifying who to punish and establishing liability, then the emphasis in accident analysis needs to shift from *cause* (in terms of events or errors), which has a limiting, blame orientation, to understanding accidents in terms of *reasons*, that is, why the events and errors occurred. In an analysis by the author of recent aerospace accidents involving software, most of the reports stopped after assigning blame—usually to the operators who interacted with the software—and never got to the root of why the accident occurred, e.g., why the operators made the errors they did and how to prevent such errors in the future (perhaps by changing the software) or why the software requirements specified unsafe behavior, why that requirements error was introduced, and why it was not detected and fixed before the software was used [116].

When trying to understand operator contributions to accidents, just as with overcoming hindsight bias, it is more helpful in learning how to prevent future accidents to focus *not* on what the operators did "wrong" but on why it made sense for them to behave that way under those conditions [51]. Most people are not malicious but are simply trying to do the best they can under the circumstances and with the information they have. Understanding why those efforts were not enough will help in changing features of the system and environment so that sincere efforts are more successful in the future. Focusing on assigning blame contributes nothing toward achieving this goal and may impede it by reducing openness during accident investigations, thereby making it more difficult to find out what really happened.

A focus on blame can also lead to a lot of finger pointing and arguments that someone or something else was more to blame. Much effort is usually spent in accident investigations on determining which factors were the most important and assigning them to categories such as root cause, primary cause, contributory cause. In general, determining the relative importance of various factors to an accident may not be useful in preventing future accidents. Haddon [77] argues, reasonably, that countermeasures to accidents should *not* be determined by the relative importance of the causal factors; instead, priority should be given to the measures that will be most effective in reducing future losses. Explanations involving events in an event chain often do not provide the information necessary to prevent future losses, and spending a lot of time determining the relative contributions of events or conditions to accidents (such as arguing about whether an event is the root cause or a contributory cause) is not productive outside the legal system. Rather, Haddon suggests that engineering effort should be devoted to identifying the factors (1) that are easiest or most feasible to change, (2) that will prevent large classes of accidents, and (3) over which we have the greatest control.

Because the goal of this book is to describe a new approach to understanding and preventing accidents rather than assigning blame, the emphasis is on identifying *all* the factors involved in an accident and understanding the relationship among these causal factors in order to provide an explanation of why the accident occurred. That explanation can then be used to generate recommendations for preventing losses in the future. Building safer systems will be more effective when we consider all causal factors, both direct and indirect. In the new approach presented in this book, there is no attempt to determine which factors are more "important" than others but rather how they all relate to each other and to the final loss event or near miss.

One final new assumption is needed to complete the foundation for future progress:

New Assumption 7: *Blame is the enemy of safety. Focus should be on understanding how the system behavior as a whole contributed to the loss and not on who or what to blame for it.*

Table 2.1
The basis for a new foundation for safety engineering

Old Assumption	New Assumption
Safety is increased by increasing system or component reliability; if components do not fail, then accidents will not occur.	High reliability is neither necessary nor sufficient for safety.
Accidents are caused by chains of directly related events. We can understand accidents and assess risk by looking at the chains of events leading to the loss.	Accidents are complex processes involving the entire sociotechnical system. Traditional event-chain models cannot describe this process adequately.
Probabilistic risk analysis based on event chains is the best way to assess and communicate safety and risk information.	Risk and safety may be best understood and communicated in ways other than probabilistic risk analysis.
Most accidents are caused by operator error. Rewarding safe behavior and punishing unsafe behavior will eliminate or reduce accidents significantly.	Operator error is a product of the environment in which it occurs. To reduce operator "error" we must change the environment in which the operator works.
Highly reliable software is safe.	Highly reliable software is not necessarily safe. Increasing software reliability will have only minimal impact on safety.
Major accidents occur from the chance simultaneous occurrence of random events.	Systems will tend to migrate toward states of higher risk. Such migration is predictable and can be prevented by appropriate system design or detected during operations using leading indicators of increasing risk.
Assigning blame is necessary to learn from and prevent accidents or incidents.	Blame is the enemy of safety. Focus should be on understanding how the system behavior as a whole contributed to the loss and not on who or what to blame for it.

We will be more successful in enhancing safety by focusing on why accidents occur rather than on blame.

Updating our assumptions about accident causation will allow us to make greater progress toward building safer systems in the twenty-first century. The old and new assumptions are summarized in table 2.1. The new assumptions provide the foundation for a new view of accident causation.

2.8 Goals for a New Accident Model

Event-based models work best for accidents where one or several components fail, leading to a system failure or hazard. Accident models and explanations involving only simple chains of failure events, however, can easily miss subtle and complex

couplings and interactions among failure events and omit entirely accidents involving no component failure at all. The event-based models developed to explain physical phenomena (which they do well) are inadequate to explain accidents involving organizational and social factors and human decisions and software design errors in highly adaptive, tightly-coupled, interactively complex sociotechnical systems—namely, those accidents related to the new factors (described in chapter 1) in the changing environment in which engineering is taking place.

The search for a new model, resulting in the accident model presented in part II, was driven by the following goals:

- *Expand accident analysis by forcing consideration of factors other than component failures and human errors.* The model should encourage a broad view of accident mechanisms, expanding the investigation from simply considering proximal events to considering the entire sociotechnical system. Such a model should include societal, regulatory, and cultural factors. While some accident reports do this well, for example the space shuttle *Challenger* report, such results appear to be ad hoc and dependent on the personalities involved in the investigation rather than being guided by the accident model itself.
- *Provide a more scientific way to model accidents that produces a better and less subjective understanding of why the accident occurred and how to prevent future ones.* Event-chain models provide little guidance in the selection of events to include in the accident explanation or the conditions to investigate. The model should provide more assistance in identifying and understanding a comprehensive set of factors involved, including the adaptations that led to the loss.
- *Include system design errors and dysfunctional system interactions.* The models used widely were created before computers and digital components and do not handle them well. In fact, many of the event-based models were developed to explain industrial accidents, such as workers falling into holes or injuring themselves during the manufacturing process, and do not fit system safety at all. A new model must be able to account for accidents arising from dysfunctional interactions among the system components.
- *Allow for and encourage new types of hazard analyses and risk assessments that go beyond component failures and can deal with the complex role software and humans are assuming in high-tech systems*. Traditional hazard analysis techniques, such as fault tree analysis and the various other types of failure analysis techniques, do not work well for human errors and for software and other system design errors. An appropriate model should suggest hazard analysis techniques to augment these failure-based methods and encourage a wider

variety of risk reduction measures than redundancy and monitoring. In addition, risk assessment is currently firmly rooted in the probabilistic analysis of failure events. Attempts to extend current probabilistic risk assessment techniques to software and other new technology, to management, and to cognitively complex human control activities have been disappointing. This way forward may lead to a dead end, but starting from a different theoretical foundation may allow significant progress in finding new, more comprehensive approaches to risk assessment for complex systems.

- *Shift the emphasis in the role of humans in accidents front errors (deviations from normative behavior) to focus on the mechanisms and factors that shape human behavior (i.e., the performance-shaping mechanisms and context in which human actions take place and decisions are made).* A new model should account for the complex role that human decisions and behavior are playing in the accidents occurring in high-tech systems and handle not simply individual decisions but also sequences of decisions and the interactions among decisions by multiple, interacting decision makers [167]. The model must include examining the possible goals and motives behind human behavior as well as the contextual factors that influenced that behavior.
- *Encourage a shift in the emphasis in accident analysis from "cause"—which has a limiting, blame orientation—to understanding accidents in terms of reasons, that is, why the events and errors occurred* [197]. Learning how to engineer safer systems is the goal here, not identifying whom to punish.
- *Examine the processes involved in accidents and not simply events and conditions* Processes control a sequence of events and describe changes and adaptations over time rather than considering events and human actions individually.
- *Allow for and encourage multiple viewpoints and multiple interpretations when appropriate* Operators, managers, and regulatory agencies may all have different views of the flawed processes underlying an accident, depending on the hierarchical level of the sociotechnical control structure from which the process is viewed. At the same time, the factual data should be separated from the interpretation of that data.
- *Assist in defining operational metrics and analyzing performance data.* Computers allow the collection of massive amounts of operational data, but analyzing that data to determine whether the system is moving toward the boundaries of safe behavior is difficult. A new accident model should provide directions for identifying appropriate safety metrics and operational auditing procedures to evaluate decisions made during design and development, to determine whether controls over hazards are adequate, to detect erroneous operational

and environmental assumptions underlying the hazard analysis and design process, to identify leading indicators and dangerous trends and changes in operations before they lead to accidents, and to identify any maladaptive system or environment changes over time that could increase accident risk to unacceptable levels.

These goals are achievable if models based on systems theory, rather than reliability theory, underlie our safety engineering activities.