Chapter

# 9

# Terminology

> *Defining concepts is frequently treated by scientists as an annoying necessity to be completed as quickly and thoughtlessly as possible. A consequence of this disinclination to define is often research carried out like surgery performed with dull instruments. The surgeon has to work harder, the patient to suffer more, and the chances for success are decreased.*
>
> —Russell L. Ackoff
> *Towards a System of Systems Concepts*

As in most new fields, terms in system safety are not used consistently. Differences exist among countries and industries. The confusion is compounded by the use of the same terms, but with different definitions, by engineering, computer science, and natural language. The goal of this chapter is to establish the definitions of a few basic terms that are used in this book—*failure, accident, hazard, risk*, and *safety*—and to differentiate safety from related qualities.

An attempt is made in this book to be consistent with engineering terminology even though the definitions may conflict with computer science usage; the goal of this book is to enhance communication and to deal with systems in general, including those containing computers. When computer scientists redefine standard engineering terms, a great deal of confusion and misunderstanding often results, which can lead indirectly to accidents or to ineffective procedures to increase safety.

## 9.1 Failure and Error

**Definition.** *Reliability* is the probability that a piece of equipment or component will perform its intended function satisfactorily for a prescribed time and under stipulated environmental conditions.

Unreliability is the probability of failure. Therefore,

**Definition.** *Failure* is the nonperformance or inability of the system or component to perform its intended function for a specified time under specified environmental conditions.

A distinction is often made between two causes of failure in physical devices. A failure may be caused by design flaws—the intended, designed and constructed behavior does not satisfy the system goal. This type of failure is sometimes called a *systemic failure*. Alternatively, a failure may result from a deviation from the originally designed behavior—the operation does not follow the original design, perhaps because of environmental disturbances or changes in the structure or design such as wear-out or degradation over time. Both of these types will be categorized as failures here and qualified (normally by denoting the mechanism behind the failure) if necessary.

**Definition.** An *error* is a design flaw or deviation from a desired or intended state.

Note that a failure is defined as an event (a behavior) while an error is a static condition (a state).[1] A failure occurs at a particular instant in time; an error remains until removed, usually through some sort of human intervention. Abstractions, models, designs, diagrams, programs, and other things that do not operate (but have states) can be erroneous, but they do not fail. Failures occur when designs are realized in concrete devices and the devices are operated. An error or erroneous state may lead to an operational failure (inability of the system to perform its expected function). A failure, in turn, may lead to an erroneous system state.

Software itself does not fail; it is a design for a machine, not a machine or a physical device. However, the computer on which the software is executing may fail, either because of problems in the computer hardware or errors in the software being executed on that hardware. Computer hardware failures may, in turn, be wear-out failures or systemic failures (resulting from computer hardware design errors). Software-related computer failures are always systemic.

Engineers distinguish between a *fault* and a *failure*, but they use the term *fault* differently than it is used in computer science. In engineering, failures are

[1] The one intentional exception to this distinction here is the use of the term *human error*. This term is too ingrained in our language and in psychology to try to make it match the

basic abnormal occurrences such as a burned-out bearing in a pump or a short circuit in an amplifier [344]. If a relay fails to close properly when a voltage is impressed across its terminals, then this event is a relay failure. Faults, on the other hand, are higher-order events. If the relay closes at the wrong time due to the improper functioning of some upstream component, then the relay has not failed but untimely relay operation may well cause the entire circuit to enter an unsatisfactory state—this event is called a fault. In general, *all failures are faults, but not all faults are failures*. For example, the relay closing when it should not is a fault. If the fault was caused by a problem within the relay itself, it is also a failure. If the valve fault was due to a spurious signal from a shorted amplifier, then this fault does not involve a failure of the valve (although the amplifier did fail).

Vesely and colleagues [344] provide another example taken from one of the earliest battles of the American Civil War. General Beauregard sent a message to one of his officers via mounted messenger #1. Some time later, the overall situation changed, and he sent out an amended message via mounted messenger #2. Still later, he sent a further amended message via mounted messenger #3. All messengers arrived, but in the wrong order. No failure occurred, but the events had a deleterious effect on the progress of the battle. This is an example of a fault that does not involve a failure.

Frequently, a distinction is also made between primary faults, secondary faults, and command faults [217]. In a *primary fault* (and failure), a component fails within the design envelope or environment. This type of failure occurs in an environment and under a loading for which the component is qualified—such as a pressure vessel bursting at less than the design pressure. Most often, this type of failure is caused by defective design, manufacture, or construction. It may also be caused by excessive or unanticipated wear or by improper maintenance and replacement policy.

*Secondary faults* (and failures) occur when components fail because of excessive environmental stresses that exceed the requirements specification or design environment. They occur in an environment or under a loading for which the component is not qualified—such as a pressure vessel failing because of excessive pressure for which it was not designed. Such failures occur randomly and are characterized by constant failure rates.

*Command faults* involve the inadvertent operation of the component because of a failure of a control element—the component operates correctly, but at the wrong time or place [217]. For example, the pressure vessel might lose pressure through the inadvertent opening of a relief valve, even though there is no excessive pressure. If the valve opened because of an erroneous signal, there was a command fault.

These types of failures and faults can be interpreted in the same way for computers. If the computer fault or failure is due to problems with the underlying hardware, then the analogies are obvious. If the computer fault is related to the software, then primary faults occur when software errors result in the computer

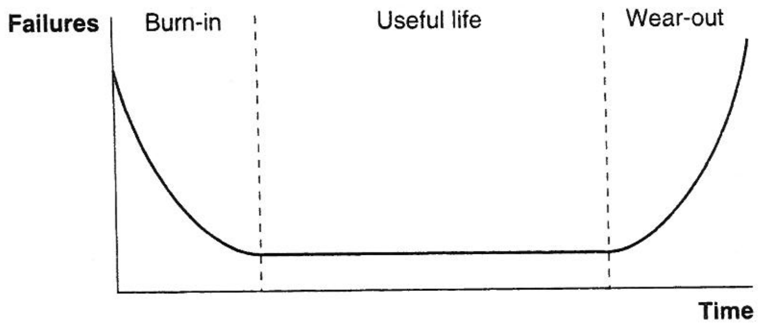

FIGURE 9.1
"Bathtub" model of reliability for electronic components, so-called because of its shape. Mechanical components tend not to exhibit the same type of constant failure rates over most of their lifetime and thus are more V-shaped.

gets inputs that differ from what was anticipated and designed for; and command faults occur when the computer responds to erroneous inputs that are expected but occur at the wrong time or in the wrong order.

A third and final distinction is often made among three different types of equipment failures (Figure 9.1):

- □ **Early failures** occur during a *debugging* or *burn-in* period and are due to poor assemblies or to weak, substandard components that fail soon after system startup. These failures are gradually eliminated, with a resulting decrease in failure rate until the failure rate reaches a fairly constant level. Software-related computer failures also exhibit early high failure rates, which decrease after testing and use in an operational environment. Early failure patterns may recur when the software is modified. Early software-related computer faults or failures are often due to incorrect assumptions about the operating environment.
- □ **Random** or **chance failures** result from complex, uncontrollable, and sometimes unknown causes. The period during which malfunctions are due primarily to random failures is the *useful life* of the component or system. The failure rate during this time is often assumed to be constant. The application of hardware reliability models to computers in order to measure operational software reliability assumes that random failures exist for software too. This assumption is based on the argument that inputs leading to computer failures or faults are encountered randomly in the input space. Note that the application of these models to the software alone makes no sense: Reliabil[illegible] when the software is executed on a particular computer; for example, differences in computer hardware can change timing and other performance characteristics.
- □ **Wearout failures** begin when the components are past their useful life: The malfunction rate increases sharply in old age. This type of failure does occur in computers, but it is due primarily to hardware failures. Some computer scientists argue that software modification and maintenance cause the computer failure rate to increase after a while, but this mechanism is not what engineers describe as wearout; instead, software maintenance errors are more closely related to early failures in hardware. In fact, every time software is modified, it can be thought of as having a new design that must undergo another burn-in period. Frequently modified software may never get beyond exhibiting early failure behavior.

## 9.2 Accident and Incident

**Definition.** An *accident* is an undesired and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss.

There are several important aspects of this definition. First, an accident is *undesired*, and because it is undesired, it is also *unplanned* or *unintentional*, although it may or may not be foreseen: What is possibly planned are prevention and remedial measures. Natural language usage of the term *accident* does sometimes imply something that is unforeseen, but engineering usage does not. We can foresee or expect automobile accidents, but they are not planned or desired. In natural language, the term *accident* can also mean something that is unavoidable, although again this meaning is not used in engineering. Many accidents are avoidable, although they may not be avoided for various reasons. The use of the qualifier *unplanned* excludes events caused by hostile action (such as sabotage).

Second, an accident results in a *specified level of loss*, which implies that there must be some type of damage to life, property, or the environment. The damage may be immediate, or it may be long term and only affect future generations. What level of loss is significant enough to be labeled an accident is subjective. Thus, what is an accident for a particular system must be defined, just as correct or expected behavior must be defined. Sometimes the definition of a specific type and level of loss is provided by the government; at other times, it is provided by the commissioner, builder, or user of the system. Sometimes distinctions are made between different levels of loss, such as catastrophic accidents, serious accidents, and minor accidents.

Finally, an accident is defined as a *loss event*, without placing limits on the type of event. Occasionally, an accident is defined in terms of causal mechanisms (for example, as a loss of control of an energy source) or a limited type of event (such as an unwanted or uncontrolled release of energy). In providing definitions of engineering terms, it is important not to limit the possible solution space of a problem by the definition itself. By defining an accident in terms of uncontrolled energy, certain types of events are excluded, such as energy deficiencies (suffocation) or toxic exposures. To include harmful exposures, the Department of Defense uses *mishap* instead of *accident*. But this new term just substitutes a different model—it defines an accident as an unwanted or uncontrolled release of energy or a toxic exposure while excluding other types of accidents that are not necessarily of concern in military systems. Accident models are discussed further in the next chapter.

The definition of an accident event is important because it influences the approach taken to increase safety. For example, if an accident is defined as an unwanted or uncontrolled release of energy, then prevention measures should focus on energy controls and barriers between the possibly harmful energy flow and the things that can be damaged by it. If the accident is defined in terms of a different underlying mechanism or model, then other approaches to preventing losses are viable. To avoid limiting solutions by the definition, an accident is defined here without any limitation on the type of loss event considered.

An incident can be differentiated from an accident.

> **Definition.** A *near miss* or *incident* is an event that involves no loss (or only minor loss) but with the potential for loss under different circumstances.

For example, a release of a toxic substance that dissipates in the air causing no harm is an incident, not an accident. It might have led to an accident given different circumstances, such as people being in the vicinity or different wind or weather conditions. Natural language is fairly imprecise about this distinction, but it is important in engineering and leads to the concept of a hazard.

## 9.3 Hazard

To prevent accidents, something must be known about their precursors, and these precursors must be under the control of the system designer. To satisfy these requirements, system safety uses the concept of a *hazard*.

A hazard has been defined in various ways. Some define it as an inherent property of an object, substance, or system that has the potential to cause harm—such as chlorine or a falling rock. Others note that the substance itself is not the hazard; rather the hazard is a set of conditions (a state) associated with that substance. Chlorine, for example, is not harmful if it is properly contained, but it may become harmful if it is released in significant quantity into the air. Similarly, water is not a hazard, but it is easy to think of combinations of conditions in which it could lead to death by drowning, scalding, or automobile accident [194].

Occasionally, a hazard is defined as an event (such as an explosion), but for various technical reasons, it will be defined as a state here. There is no significant difference, since states can be thought of as leading to events, which in turn create new states. Thus, there is no important difference in defining the hazard as, say, the system exploding or as the system having explosive energy.

The problem with the usual definition of a hazard—something that has the potential to do harm or that can lead to an accident—is that most every state, given certain conditions, has the potential to do harm or can lead to an accident. An airplane that is in the air is in a hazardous state, but there is little that the designer of an air traffic control system, for example, can do about designing a system where planes never leave the ground.

Considering these factors, the definition to be used here is

> **Definition.** A *hazard* is a state or set of conditions of a system (or an object) that, together with other conditions in the environment of the system (or object), will lead inevitably to an accident (loss event).

There are some things to note about this definition.

- *A hazard is defined with respect to the environment of the system or component.*

In most cases, accidents involve the environment within which a component or system exists. As an example, the release of toxic material or explosive energy will cause a loss only if there are people or structures in the vicinity. Weather conditions may also affect whether a loss occurs in the case of a toxic release. If the appropriate environmental conditions do not exist, then there is no loss and, by definition, no accident.

The only exception is a physical system where the boundaries have been drawn such that they include the object that is damaged plus all the conditions necessary for the loss. Note that, by definition, the latter case cannot happen for software since it is not a physical object, only an abstraction. Thus, software by itself is not safe or unsafe, although it could theoretically become unsafe when executed on a computer. But even then, there are few hazards that are inherent in the computer system itself since computers do little besides generate electronic signals. They can catch fire or fall on someone, but these hazards normally have nothing to do with the software design. Thus, we can only talk about the safety of software and its hazards in the context of the particular system design within which it is being used. Otherwise, the hazards associated with software do not exist. That there are no inherent software hazards is one of the reasons that many system safety engineers prefer the term *software system safety* to *software safety*.

- *What constitutes a hazard depends upon where the boundaries of the system are drawn.*

A system is an abstraction, and therefore the boundaries of the system can be drawn anywhere the person who is defining the system wants. The boundaries, in turn, determine which conditions are considered part of the hazard and which are considered part of the environment. Since this choice is arbitrary, it is most useful to define the boundaries (and thus the hazard) in such a way that safeguards can be implemented within the constraints of also achieving the basic mission and other system goals: The system boundaries should be drawn to include the conditions related to an accident over which the system designer has some control. At the extreme, they can be drawn to include all conditions involved in the accident, but drawing the boundaries in this way would serve no purpose since many of these conditions are not controllable by the designer. Normally, we try to define a system we are building in such a way that we have control over the states.

Sometimes, an accident is defined as the non-accomplishment of the system mission or the loss of the system itself, such as the loss of a spacecraft. Even in these cases, the loss may involve environmental variables (for example, electromagnetic particles) over which the designer has little control beyond attempting to shield the system against them.

As an example of how to define hazards and system boundaries, consider an air traffic control system. If an accident is defined as a collision between two aircraft, then an appropriate hazard is the lack of minimum separation between aircraft. The designer of a collision avoidance system or a more general air traffic control system theoretically has control over the separation between aircraft, but may not have control over other factors that determine whether two aircraft that get close together actually collide (such as the weather conditions or the state of mind or attentiveness of the pilots). As noted earlier, a hazard can be defined as two planes being in the same air space, but this definition is not useful as the state is inevitable and cannot be avoided. For practical reasons, we need to define hazards as the states we want to avoid.

As another example, for flammable mixtures to catch fire or explode, there must be both air and a source of ignition. Kletz argues that when flammable gases or vapors are handled on an industrial scale and mixed with air in flammable concentrations, experience shows that sources of ignition are likely to turn up [154]. Therefore, the only safe rule is to assume that mixtures of flammable vapor in air in the explosive range will sooner or later catch fire or explode and should never be deliberately permitted, except under carefully defined circumstances where the risk is accepted. Using this argument, the hazard might be defined as a mixture of vapor in air (and not the ignition source), since those are the only two of the three necessary conditions over which control can be exercised.

In summary, the definition of a hazard is arbitrary, and one of the first steps in designing a system is to decide what conditions will be considered to be hazards that need to be eliminated or controlled.

Occasionally, it is useful to classify hazards as *endogenous* or *exogenous* [208]. An endogenous hazard is caused by defects in design, material, workmanship, or operating procedures—that is, by factors inherent in the system or device itself. In contrast, an exogenous hazard is brought about by phenomena external to the system, such as lightning or cosmic radiation.

A hazard has two important characteristics: (1) *severity* (sometimes called *damage*) and (2) *likelihood* of occurrence. Hazard *severity* is defined as the worst possible accident that could result from the hazard given the environment in its most unfavorable state.

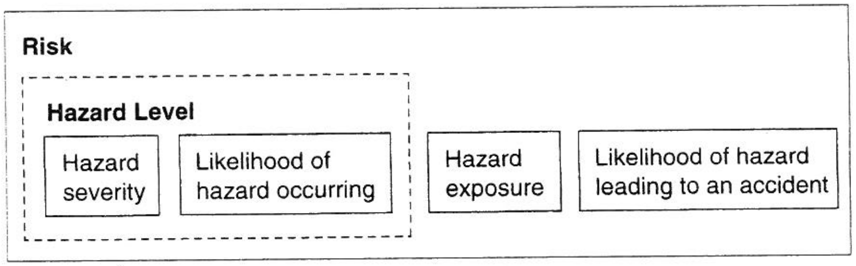
Risk
Hazard Level
Hazard severity
Likelihood of hazard occurring
Hazard exposure
Likelihood of hazard leading to an accident

FIGURE 9.2
The components of risk.

The hazard *likelihood* of occurrence can be specified either qualitatively or quantitatively. Unfortunately, when the system is being designed and hazards are being evaluated for potential tradeoffs and ranked as to which should be eliminated first, the information needed to evaluate the likelihood accurately is almost never available. For a few hazards associated with standard designs, historical data is available. For the rest, qualitative evaluation of likelihood is usually the best that can be done.

The combination of severity and likelihood of occurrence is often called the *hazard level*. Hazard level, along with two other factors related to hazards, are used in the definition of risk.

## 9.4 Risk

**Definition.** *Risk* is the *hazard level* combined with (1) the likelihood of the hazard leading to an accident (sometimes called *danger*) and (2) hazard exposure or duration (sometimes called *latency).*

Sometimes risk is limited to the relationship between the hazard and the accident (the likelihood of the hazard leading to an accident but not the likelihood of the hazard occurring). In this book, the more inclusive definition is used (Figure 9.2).

*Exposure* or *duration* of a hazard is a component of risk: Since an accident involves a coincidence of conditions, of which the hazard is just one, the longer the hazardous state exists, the greater the chance that the other prerequisite conditions will occur. The coincidence of conditions necessary for an accident may be a statistically low-probability event, but the probability of coincidence can be dramatically increased if the hazard is present over long periods [274].

As an example of the definition of risk, if a computer has a control function such as controlling the movement of a robot, a very simple model [61] defines risk as a function of the

1. Probability the computer causes a spurious or unexpected machine movement
2. Probability a human is in the field of movement
3. Probability the human has no time to move or will fail to diagnose the robot failure
4. Severity of worst-case consequences

If the computer has a continuous protective or monitoring function, along with a requirement to initiate some safety function upon detection of a potentially hazardous condition, then another example risk definition is a function of the

1. Probability of a dangerous plant condition arising
2. Probability of the computer not detecting it
3. Probability of the computer not initiating its safety function
4. Probability of the safety function not preventing the hazard
5. Probability of conditions occurring that will cause the hazard to lead to an accident
6. Worst-case severity of the accident

If it is assumed that all of these events are independent, the probabilities could be multiplied together, but this assumption is normally not realistic and a more complex relationship and computation are required. For example, the probability of a person being in the field of movement of a robot may be higher if the robot is behaving strangely—the operator may have approached in order to investigate. A more sophisticated model also would include such factors as the exposure time of the hazard (the average time to detection and repair).

In almost all cases, the correct way to combine the elements of the risk function is unknown, as are the values of the parameters of the function. In addition, agreement has not been reached on how to combine probability and severity and other nonprobabilistic factors such as exposure time. Finally, how can an event that is catastrophic but very unlikely be compared with another event that is much more likely but less serious? Ad hoc quantitative methods could be devised to make this comparison, but, in the end, the process must necessarily involve qualitative judgment and personal values and is therefore trans-scientific.

Sometimes the terms *risk analysis* and *hazard analysis* are used interchangeably, but an important distinction exists. *Hazard analysis* (as defined here) involves only the identification of hazards and the assessment of hazard level, while *risk analysis* adds the identification and assessment of the environmental conditions along with exposure or duration. Thus, hazard analysis is a subset of risk analysis.

As has been discussed earlier, risk (or safety) is sometimes confused with reliability, and reliability measurement is often used incorrectly as a measure of risk. Reliability is a component property, whereas safety or risk cannot be defined or measured without considering the environment. As an example, we can talk about the reliability of a pistol (the probability that it will fire when the trigger is pulled), but to talk about the "risk of the pistol" is meaningless by itself. Consider a pistol being fired in the middle of an uninhabited forest versus being fired in the middle of a crowded shopping mall. The reliability in each situation has not changed and neither has the pistol, but the risk or safety of the two situations is very different. In fact, if the reliability of the pistol is relatively high (as it usually is), it becomes an almost inconsequential factor in assessing the risk in these two situations; reliability, in this case, is swamped by the other factors involved in calculating the risk of injury.

While it is indisputably true that reliability is a factor in safety or risk and thus should be included in risk assessments, other factors may be equally or even more important. However, because component failure is the most convenient thing to measure, we often use it as *the* measure of risk or assign it too much importance in risk assessments. Most accidents in complex systems involve factors other than single component failure.

## 9.5 Safety

Safety is defined in this book in an absolute sense:

**Definition.** *Safety* is freedom from accidents or losses.

Some people have argued that there is no such thing as absolute safety, and therefore safety should be defined in terms of acceptable loss. William Lowrance is usually credited with originating this alternative definition: "We will define safety as a judgment of the acceptability of risk, and risk, in turn, as a measure of the probability and severity of harm to human health. A thing is safe if its attendant risks are judged to be acceptable" [197].

Lowrance himself raises questions about this definition: What is meant by "acceptable" risk? To whom is the risk posed? By whom is it judged acceptable? A condition that is acceptable to an employer may not be acceptable to the employee and vice versa. These questions lead to endless arguments about what level or type of loss is "acceptable."

One can envision safety, like other qualities, along a continuum, with one end being freedom from losses and the continuum stretching toward increasing loss (Figure 9.3). If "safe" is not at the left end of the continuum, then where should it be? That question is trans-scientific, as argued in Chapter 1, and, in general, definitions of engineering terms should not involve trans-scientific concepts. To avoid the hopeless quagmire of arguments about acceptability, it is simplest to put safety at the left end of the continuum and then determine how close one comes to that ideal.

In fact, many qualities are ideals that can only be approached asymptotically. There is no such thing as a totally secure system; anything can be compromised.

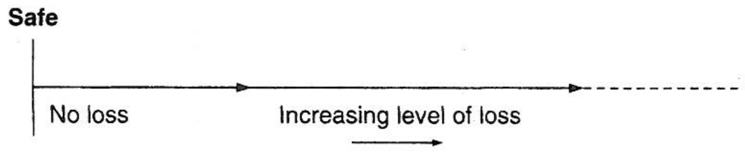

FIGURE 9.3
Safety as a continuum.

Nevertheless, that does not keep us from defining security in absolute terms. The same is true for reliability—nearly everything will break or wear out over time or under some conditions. We deal with this dilemma by defining reliability very narrowly—that is, by restricting the conditions and time under which we evaluate reliability in a particular system. The same could be done for safety; we could restrict the definition to identified hazards and conditions. But those who suffer losses are unlikely to accept that an accident did not happen (the system was safe) despite the loss, just because the specified conditions or time were exceeded or an unidentified hazard was involved.

A relative definition of safety also implies that hazards cannot be eliminated, when they often can. While, in most instances, *all* hazards cannot be eliminated, this book describes many ways that specific hazards can be totally eliminated from a product or system. Often, hazard elimination requires sacrificing some other goals or requires more knowledge and effort, especially up-front design effort, but it is not impossible. Thus, it *does* make sense to talk about absolute safety from a particular hazard. By accepting a relative definition of safety, it is possible to ignore design alternatives that eliminate or greatly reduce particular hazards but require compromises with respect to other goals.

## 9.6 Safety and Security

Arguments have been advanced that safety is a subset of reliability or a subset of security or a part of human engineering—usually by people in these fields. Although there are some commonalities and interactions, safety is a distinct quality and should be treated separately from other qualities, or the tradeoffs, which are often required, will be hidden from view.

Safety and security, however, are closely related, and their similarities can be used to the advantage of both in terms of borrowing effective techniques from each to deal with the other. Both qualities deal with threats or risks—one with threats to life or property and the other with threats to privacy or national security. Both often involve negative requirements or *constraints* that may conflict with some important system goals. Both involve protection against losses, although the types of losses involved may be different. Both involve global system properties that are difficult to deal with outside of a system context. Both involve requirements that are considered of supreme importance (in relation to other requirements) in deciding whether the system can and should be used—that is, particularly high levels of assurance may be needed, and testing alone is insufficient to establish those levels [170]. In fact, a higher level of assurance that a system is safe or secure may be needed than that the system performs its intended function. Finally, both qualities involve aspects of a system that are regulated by government agencies or licensing bureaus (such as the National Security Agency and the Nuclear Regulatory Commission), where approval is based on factors other than whether the system does anything useful or is economically profitable.

These shared characteristics lead to other similarities. Both may benefit from using technologies that are too costly to be applied to the system as a whole, such as formal verification, but that may be cost-effective for these limited subsets of the requirements. Both also involve problems and techniques that apply specifically to them and not to other, more general functional requirements or constraints.

Some of the techniques applicable to one are applicable to the other. For example, both can benefit from the use of barriers. For security, barriers are used to prevent malicious incursions rather than accidental ones, but the technique is the same. Other security techniques do not seem to apply to safety—for example, the use of traps to encourage attacks against hidden defenses or the randomization of limited defensive resources to reduce the expected success of planned attacks [98].

There are also important differences between safety and security. Security focuses on malicious actions, whereas safety is also concerned with well-intended actions. In addition, the primary emphasis in security traditionally has been on preventing unauthorized access to classified information, as opposed to preventing more general malicious activities. Note, however, that if an accident or loss event is defined to include unauthorized disclosure, modification, and withholding of data, then security becomes a subset of safety.

The definition of safety or security could be extended to include both qualities, but nothing appears to be gained by making this extension, while important differences become obscured. Separation of qualities to better control and understand them, to allocate limited resources, and to enforce priorities is an appropriate goal. However, attempts to integrate several qualities into one abstraction (like *dependability*, which has been proposed as a combined measure of reliability, safety, security, availability and just about every other quality) seem misguided. These global abstractions have only disadvantages, since they inhibit understanding and control. Often qualities conflict, and *de facto* tradeoffs are lost in global abstractions or measurements. For example, it is possible to increase dependability while decreasing safety without it being at all apparent that this increase in risk has occurred.

## 9.7 Summary

This chapter has tried to clarify some basic concepts and establish workable definitions. Agreeing on terminology is always a difficult process, but it is important for communication and progress in finding solutions to our problems: Definitions can have powerful effects on the way we express our problems and therefore how we go about solving them. Establishing a common terminology is always painful but is worth the effort in the long run.

# Chapter 10

# Accident and Human Error Models

*Accidents on the whole are becoming less and less attributable to a single cause, more to a number of contributory factors. This is the result of the skill of the designers in anticipating trouble, but it means that when trouble does occur, it is inevitably complicated.*

—DeHavilland and Walker<br>
(after reviewing failures<br>
of the Comet aircraft)[1]

Models provide a means of understanding complex phenomena and record that understanding in a way that can be communicated to others. All models abstractions—they simplify our world by abstracting away irrelevant details focusing on the features that are assumed to be the most relevant.

The design and analysis methods used for safety-critical systems are b on particular underlying models of the accident process and of human er How effective our procedures are depends, to a large extent, on how accu our models are—that is, how well they reflect the features of the environ to which they are applied. To design an effective safety program and sele appropriate set of procedures and techniques, we need to understand the m that underlie our options and the assumptions about accidents and human e they embody.

[1] Quoted in [339].