Chapter 10
Errors: An Inherent Part of Human-System Performance

Abstract In this chapter we consider how errors contribute to accidents, large and small, and what we can do about them. We discuss the problem of post-hoc analyses, the types of human error that can occur, and how to design systems in such a way that the errors can be appropriately managed. The examples illustrate how user's characteristics in terms of psycho-physiology, fatigue, cognitive processing, and social situations can all contribute to failures. We especially note the importance of Norman's (and others') main guideline about needing to design for error.

10.1 Introduction to Errors

In this chapter we provide an introduction to the topic of what is often called *human error*. As Reason (1990) notes, "human error is a very large subject, quite as extensive as that covered by the term human performance," so we can only really provide a selective overview of some of the major issues.

We have deliberately separated the discussion of errors into a separate chapter. This is because errors are an inherent part of system performance. In other words, they often arise as a combination of factors at the anthropomorphic, behavioral, cognitive, and social levels in the ABCS framework. If you look again at the example of the Kegworth air accident (see the Appendix), you should be able to appreciate this more fully at this point in the book.

Our purpose here is to highlight the need to think about your users in context, and to determine what kinds of factors can give rise to erroneous performance. By highlighting the relationship between system performance and error, we hope to show you why it is important to think about designing for error (Norman 1988, 2013). One way of designing for error is to identify the situations that can lead to erroneous performance, and then put in place appropriate mechanisms to either prevent the errors, or at least mitigate the adverse consequences arising from those errors.

F. E. Ritter et al., *Foundations for Designing User-Centered Systems*,
DOI: 10.1007/978-1-4471-5134-0_10, © Springer-Verlag London 2014

We will illustrate our points using examples taken from a range of incidents and accidents, large and small. In doing so, we hope to show how errors do not just arise because of any inherent error-proneness or maliciousness of the users. Instead, errors are usually the result of an interaction of several contributing factors (people, technological, and contextual). Once we accept this state of affairs we can begin to move away from the need to find someone to blame, and start to learn from erroneous performance as a way of improving future system performance.

10.1.1 What is Error?

Errors are generally regarded as precursors to accidents. The error triggers a set of events—often referred to as a chain or sequence, although it is not always a linear set of events—ultimately leading to an outcome that has serious consequences involving significant loss of life, money, or machinery. Causal analyses of accidents usually highlight the fact that there were many contributory factors. There are obviously exceptions, where a single catastrophic failure leads directly to an accident, but generally accidents involve a series of several individually minor events. This process is sometimes described as a domino effect, or represented by the Reason's (1990) Swiss cheese model in which there are holes in the various layers of the system, and an accident only occurs when the holes line up across all the layers.

A similar idea is encapsulated in Randell's (2000) *fault-error-failure* model that comes from the field of dependability. A failure is defined as something that occurs when the service that is delivered is judged to have deviated from its specification. An error is taken to be the part of the system state that may lead to a subsequent failure, and the adjudged cause of the error is defined as a fault.

It is very important to note that identifying whether something is a fault, error, or failure involves making judgments. The fault-error-failure triples can link up so that you effectively end up with a chain of triples. This is possible because a failure at one level in the system may constitute a fault at another level. This does not mean that errors inevitably lead to failures, however. The link between an error and a failure can be broken either by chance or by taking appropriate design steps to contain the errors and their effects.

Those errors that have immediate (or near-immediate) effects on system performance are sometimes called active errors (Reason 1990). This is to distinguish them from latent errors, which can lie dormant within a system for some considerable time without having any adverse effect on system performance. The commission that investigated the nuclear accident at Three Mile Island, for example, found that an error that had occurred during maintenance (and hence was latent in the system) led to the emergency feed water system being unavailable (Kemeny (chairman) 1979). Similarly, the vulnerability of the O-ring seals on the Challenger Space Shuttle was known about beforehand and hence latent in the

system (Vaughan 1997). The vulnerability was only exploited after the decision was made to launch the shuttle in very cold weather.

The general approach used to study errors focuses on how to understand errors so that we can take appropriate steps to manage them before things get out of hand. When, rather than if, things go wrong, the aim is to learn from what happened to help prevent a repeat performance.

Immediately after a major accident of any kind the press coverage almost invariably attributes the cause of the accident to human error. The problem is that the term *human error* is ambiguous and has three subtly different meanings (Hollnagel 1998), which are often confused by the press. It is therefore worth spelling out these different meanings, using examples from the field of aviation:

1. Human error is the cause of the event or action. An example of this would be if an aircraft deviated from its assigned flight altitude to a different flight altitude (either higher or lower than the one that the flight crew had been given by Air Traffic Control) due to the actions of the flight crew.
2. Human error is the event or action itself. An example of this would be if an aircraft pilot did not change the altimeter setting when they were supposed to. Note that in this case the action is really a deliberate non-action.
3. Human error is the consequence of the event or action. An example of this would be if an aircraft collided with another aircraft because the pilot started to taxi before receiving clearance to taxi by air traffic control.

The differences between the meanings are quite subtle, but it is important to ensure that you understand them. In most cases the press combines the first two meanings, even though they may not intend that the primary attribution of blame should fall on the human.

The interpretation of human error is further complicated by the fact that an action can only be judged as erroneous in hindsight (Woods et al. 1994). People will generally do what they think is the right thing in that particular context at the right time. So an action can only be judged as being erroneous after the fact, based on:

- A comparison with some expected level of performance
- A degradation in performance
- The person who performed the act having been unable to choose to act in a way that would not have been considered as erroneous.

There is one exception to this notion of erroneous actions being judgments made in hindsight: violations. If a person deliberately decides to do the wrong thing—to sabotage the system, for example—then this can be determined at the point when the action occurs, rather than afterwards. In some cases, however, it may be necessary to violate the established rules or procedures to keep a system in a safe state, or to get it out of an unsafe one. The Federal Aviation Authority (FAA) in the US acknowledges this type of violation—sometimes called safe violations—and explicitly allows them under its regulations in certain situations.

Erroneous actions, then, can be seen generally as the result of one of two things:

1. Performing the right action in the wrong circumstances. This is what Reason (1990) calls a mistake, or a failure in planning.
2. Performing the wrong action in the right circumstances. This is often referred to as a slip (Norman 1981; Reason 1990), or failure in action execution.

In either case the action could have been deemed correct if the circumstances had been slightly different. This helps to explain Rasmussen's (1988) description of erroneous actions as being the results of carrying unsuccessful experiments in unfriendly environments.

It is also important to take into account different perceptions when trying to interpret what people mean by error. Rasmussen et al. (1994) suggest that the following perspectives can be identified:

- Common sense: to explain an unusual event
- The lawyer: to find somebody to blame and/or punish
- The therapist: to improve human performance
- The scientist: to understand human behavior
- The reliability analyst: to evaluate human performance
- The designer: to improve system configuration.

As interactive system designers, our perspective tends to be mostly a combination of the scientist's and the designer's perspectives. The two are somewhat related, because by understanding the human behavior, we can provide appropriate support to prevent errors happening or to mitigate the consequences of any errors that may not be preventable. In most cases this will be by changing the design of the system.

10.1.2 The Fine Line Between Success and Failure

As long as there have been people, there have been errors. Getting things right is not always easy, and often requires knowledge and skills that have to be acquired over an extended period of time: you cannot become an expert overnight. One of the ways in which people learn is through practice, by reflecting on their performance, using feedback, and then trying to do it better next time. This approach is nicely illustrated in the works of Henry Petroski (e.g., Petroski 1985/1992, 1994, 2006), who has shown how the development of engineering has progressed over the centuries by learning from past errors.

The study of errors has fascinated psychologists for over a century. It received renewed impetus from the end of the 1970s with major events like the Three Mile Island disaster in 1979, the runway collision at Tenerife in 1977, and a range of catastrophes in medical care (e.g., Bogner 2004), when the gauntlet was picked up by the human factors and ergonomics community. The focus of study has changed somewhat, however, and there is now recognition that it is important to think about

success as much as failure because there is often only a fine line dividing the two, and there are many more instances of success than of failure (e.g., Hollnagel et al. 2006). This changed emphasis is a reflection of Ernst Mach's (1905) prescient view that "Knowledge and error flow from the same mental sources, only success can tell one from the other……the same mental functions, operating under the same rules, in one case lead to knowledge, and in another, to error….". Although couched in slightly different terms, it is a view that others have concurred with and reiterated, such as Reason (1990, p.1):

> Not only must more effective methods of predicting and reducing dangerous errors emerge from a better understanding of mental processes, it has also become increasingly apparent that such theorizing, if it is to provide an adequate picture of cognitive control processes, must explain not only correct performance but also the more predictable varieties of human fallibility. Far from being rooted in irrational or maladaptive tendencies, these recurrent error forms have their origins in fundamentally useful psychological processes.

The consequences of errors have also increased over the years. Or perhaps that should be the consequences are *perceived* to have increased. The mass media these days are often very quick to report on air accidents, for example, where a single accident may give rise to hundreds of casualties. The fact that more people get killed on the roads, however, goes largely unreported, mostly because each fatal road accident often only involves a few deaths (Gigerenzer 2004).

10.1.3 The Accident was Caused by Human Error, Right?

Most accidents could naively be attributed to human error because the systems that fail, leading to the accident, are designed by humans. This is an over-simplistic view, however, and would lead to an equally simplistic solution, i.e., that removing the human would remove a major source of failures, and hence eliminate many accidents. The idea of humans being accident prone in a volitional way (i.e., of their own free will) dominated early thinking in human error research.

The cause of an accident is often attributed to human error (pilot error, driver error, operator error, and so on). This is a judgment that is built on several underlying assumptions that, at best, usually only represent a partial view of the true situation. There are many examples of such a view. Arnstein (1997), for example, found that the number of problems that could be attributed to human error in anesthetics ranged from 64 to 83%, whilst in aviation the range was 40–88%. Johnson and Holloway (2007) also noted the tendency to over-emphasize human error as the reported cause in transportation accidents for the years 1996–2003. Whilst it was still found to be the main attributed cause, the levels were somewhat lower at 37% for the US National Transportation Safety Board (NTSB), and 50% for the Canadian TSB.

One of the main reasons that accidents end up being attributed to human error is because of the limitations of causal analysis. It is difficult to continue the analysis through the human when we do not have direct access to what was going on in the

operators' heads when the accident happened. So once the human is reached in the sequence of attributable causes, the analysis frequently gets terminated, and we are left with human error as the result.

As we have noted several times, system performance is the result of people interacting with technology in a particular context (organizational and physical environment). The importance of context should never be underestimated. Very often, when we look at accidents we find that the users were working in a context constrained by time pressures and limited resources.

Nowadays, there is a greater awareness of the influence of the context in which work takes place, and the fact that human attentional resources are limited (see Chap. 5). Many events that previously were typically attributed to humans being accident prone would now be analyzed and categorized differently.

In aviation, for example, where multitasking is an inherent part of flying a plane, distractions are recognized as being a particular problem. Dismukes et al. (1998) noted that nearly half the reported NTSB accidents attributed to crew error involved lapses of attention associated with interruptions, distractions, or an excessive preoccupation with one task to the exclusion of another that had to be performed within a similar time frame. The vast majority (90%) of competing activities that distracted or preoccupied pilots fell into four categories: communication; head-down work; searching for other traffic in good weather (visual meteorological conditions or VMC); or responding to abnormal situations. Flight crews have to work as a team, but this has to be done in such a way that it does not detract from the individual tasks that have to be performed as the following excerpt from incident report #360761 from NASA's Aviation Safety Reporting System (ASRS) illustrates:

> Copilot was a new hire and new in type: first line flight out of training IOE. Copilot was hand-flying the aircraft on CIVET arrival to LAX. I was talking to him about the arrival and overloaded him. As we approached 12,000 feet (our next assigned altitude) he did not level off even under direction from me. We descended 400 feet before he could recover. I did not realize that the speed brakes were extended, which contributed to the slow altitude recovery.

Here the Pilot Not Flying (PNF) was trying to help the co-pilot (the Pilot Flying or PF), which led to problems on two levels. First, the combination of flying the plane and trying to heed the PNF's advice simply overloaded the PF. Second, the fact that the PNF was focused on making sure that he gave the PF appropriate assistance meant that he was distracted from his task of monitoring the ongoing status of the plane. Both flight crew members were trying to do the right thing, but they did not have enough resources to accomplish everything they needed to do. The distractions in this case were at least partly self-created; such distractions often lead to incidents in many domains (Baxter 2000). This incident would traditionally have been attributed to pilot error, but a closer examination of the context suggests that this is an over-simplification.

We noted earlier that there is a fine line between success and failure. In the aviation incident described above, where the plane descended too far, it seems

obvious that there were time pressures involved. The PF knew that he had to respond relatively quickly whilst still listening to the PNF's advice. Perhaps if the PF had been more experienced, and the PNF had not felt the need to talk the PF through the arrival route, neither of them would have been distracted from the task at hand: the PF would have been more likely to level off at the appropriate altitude, and the PNF more likely to detect any anomalies in the plane's status.

The air accident at Kegworth in the UK (Air Accidents Investigation Branch 1989), described in the Appendix offers another example of a failure that was officially attributed to pilot (human) error. One of the problems was the fact that when the crew shut down the good engine, this coincided with a reduction in vibration, and a cessation of the smoke and fumes from the faulty engine. This led the crew to believe that they had taken the correct action, which can be at least partly attributed to the use of a flawed mental model (Besnard et al. 2004).

It is also important to ensure that appropriate account is taken of the physiological limitations of users as well as their psychological limitations. A simple example is the original design of packaging for medication tablets (and some other potentially hazardous household items such as domestic bleach). It used to be quite easy for a young child to unscrew the cap from a medicine bottle and then eat the contents because they looked like sweets. The solution was the child-proof safety cap. Although children could not open them in tests, older people also found it difficult to open the cap, particularly if they suffered from arthritis. In complex situations, such as flying an airplane (and particularly smaller ones), the issues involved may be more subtle. Here it is important that the pilot is not asked to do things like move their head in one direction whilst the aircraft is moving in another, because this can lead to severe disorientation.

The design limitations of the system also need to be taken into account. What often happens is that there is a general expectation that the human operator should compensate for any inadequacies in system design. Usually training is used to bridge the gap, but sometimes users are simply left to work it out for themselves.

The technique of Crew—originally *Cockpit*—Resource Management (CRM) was developed (Wiener et al. 1993) to anticipate some potential problems that can arise from the interactions between people, technology, and context within aviation. CRM aims to minimize the potential for failures in interpersonal communications at crucial times during a flight, for example, which can lead to real accidents such as:

- A plane crashing on take-off because the distracted crew failed to complete a safety checklist that would have confirmed that the aircraft's flaps had not been extended.
- A plane crashing into a river when the co-pilot failed to get the attention of the Captain about concerns that the take-off thrust had not been properly set. The co-pilot felt that he could not tell his superior what to do.
- A plane crashing when it ran out of fuel due to a communications breakdown between the Captain, the co-pilot, and air traffic control about the amount of fuel onboard.

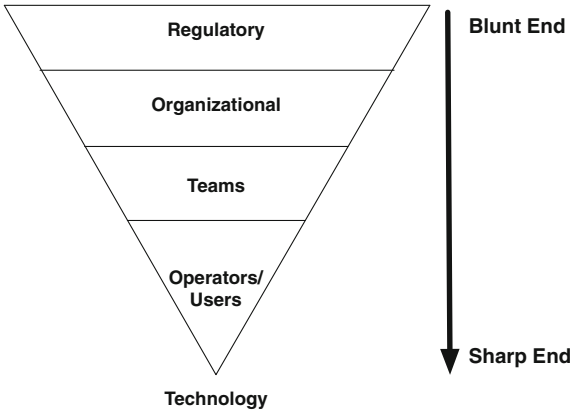

Fig. 10.1 The influence of the blunt end on the sharp end of an incident

It is only when you start to give deeper consideration to the circumstances in which the error occurred that you really appreciate how hard it is to decide who (or what) is *really* to blame. If a doctor makes a medication error after having worked for 24 h continuously, for example, can we *really* blame the doctor? We know that fatigue adversely affects cognitive performance, yet the system still put the doctor in a situation where they were likely to suffer from fatigue. Similarly, we know that the way that system controls are laid out can either help or hinder performance. If a design is inappropriate, can we really blame the users?

In fact, the actions that are performed at the lowest level (usually where the user interacts with the technology) are only a small part of the picture. This level of interaction is often referred to as *the sharp end* of the system. Actions at the sharp end are often influenced by what happens at the so-called *blunt end* of the system, as shown in Fig. 10.1. The idea of a sharp end and a blunt end comes from Woods et al. (1994). This figure illustrates that final users are often seen as causes at where users meet the task, but that there is a lot of structure behind them that influences the situation as well; structure that is harder to change but that has a large amount of influence.

Decisions and actions taken at the regulatory level can affect what the operators do. In the USA, for example, the FAA's regulations state that pilots must not perform deliberate violations, unless the violation is needed to put the aircraft into a safe state. Similarly, standard operational procedures (which are generally defined at the organizational level) usually define what the operators can (or cannot) do.

10.2 Studying Error

The issue of data collection is fundamental to the study of human error. If we could reliably predict exactly when an error was going to occur, it would be a simple matter to warn the person or people involved so that they could avoid it.

Alternatively, we could design the system so that it would prevent the error occurring, or at least mitigate the consequences of the error.

There is a standard set of questions that applies to the collection of human performance data:

- Why gather data?
- What sort of data to gather?
- Where (and when) to gather data?
- How much data to gather?
- How to gather data?

These questions provide a basic framework for the discussion of the various issues involved. The issues interact and overlap, so it is not possible to answer each of the questions in isolation.

It is worth re-emphasizing at this point that erroneous behavior is an inherent part of human performance, i.e., there is a close link between knowledge and error (Mach 1905). The corollary of this is that knowledge or data relating to correct performance are also needed to make an informed judgment regarding each potential instance of state misinterpretation.

Here we only address the question of how to gather data, focusing on complex domains where the system can change dynamically without human intervention (such as aircraft, cars, and power plants). There are three basic data collection methods that can be used: laboratory-based experiments; field-based observation; and archive data. Each has its own set of strengths and weaknesses, as discussed below. The final choice of method depends on the particular situation at hand.

10.2.1 Laboratory-Based Experiments

The first method is the standard behavioral science method of using laboratory-based experiments. The main advantage of this method is that it allows for the independent variables associated with a particular phenomenon to be experimentally controlled. By varying one (or more) independent variables, the effect on the dependent variable can be observed.

The main drawback is the lack of face validity between laboratory-based experiments and the real world situation. This lack of validity makes it inappropriate at best, and impossible at worst, to generalize from the results obtained in the laboratory to the situation in the real world. The use of laboratory-based experiments largely ignores the current consensus of opinion on the importance of context in shaping human performance (Hollnagel 1993a; Hutchins 1995; Nardi 1996).

It is a long and difficult task to develop and conduct laboratory experiments that would meet all the requirements of a situation that would definitely lead to a human error. This is partly because of the problems of availability and selection of appropriate experimental subjects. The subjects would need lengthy experience of

operating the system being used. Whilst it might be possible to use operators from the relevant domain, their experience is liable to be biased towards the particular system they normally work with (as opposed to your new system). The practicality of getting access to operators for the length of time needed to conduct the experiments also mitigates against using this approach. In addition, if subjects can identify the purpose of the experiment they may behave more cautiously, to guard against performing erroneous actions.

It can also be difficult to choose an appropriate experimental task and setting. Complex systems often require operators to perform multiple tasks, sometimes simultaneously. The difficulty is to find a complex system (or an appropriate simulation) that can be readily deployed under laboratory conditions. The system would also need to be familiar to the subjects to fulfill the criterion regarding expertise. Unfortunately, laboratories that have their own high fidelity simulations, such as Halden's nuclear power plant simulator (Hollnagel et al. 1996) are still the exception rather than the rule.

10.2.2 Field-Based Observation

The second method is to carry out longitudinal observation of experienced operators. The main advantage of this method is that it guarantees the ecological validity of the data. In general, observation is a valid technique, particularly if the aim is to investigate human reliability per se. In light of Mach's (1905) observation about the relationship between knowledge and error, there is a lot to be learned about operator performance under abnormal system operating conditions from observing performance under normal operating conditions.

Observational research tends to focus on one specific aspect of operator behavior, however, rather than on operator behavior per se. The observational method, therefore, has a number of drawbacks. The first is the relatively low frequency of occurrence of human error. Although there may be a deterministic element to the occurrence of human error, it is very difficult to predict precisely when a set of events or actions giving rise to an observable error will occur. So there is no guarantee that human error will occur, even during extended periods of observation.

The second drawback is the high costs associated with extended observation and the subsequent data analysis. Even if it could be guaranteed that human error would occur once in every 24 h period, for example, then gathering enough data for 100 errors would require 2,400 h of recordings. Because analysis of recorded data takes an order (or two) of magnitude longer than the recording, the time to gather and to analyze such a large amount of data quickly becomes prohibitive.

The third drawback is the inherent adaptability of human behavior. One of the characteristics of experienced operators is their ability to detect and recover from potential erroneous actions. In other words, recovery is performed before unwanted consequences arise. Unless verbal reports are also recorded, which can

be used to try and identify these recoveries, it can be difficult to detect where a recovery has taken place.

The fourth drawback is that the presence of an outside observer may affect the operator's behavior. In particular, operators may be more careful than usual if they know their performance will be recorded for subsequent analysis. This may reduce the frequency of occurrence of human error, thereby requiring even longer periods of observation to gather enough data.

10.2.3 Archive Data

The third method is to use existing available data. Archive accident and incident reports, for example, have been successfully used within human error research (e.g., Pew et al. 1981; Rasmussen 1980; Wagenaar and Groeneweg 1987; Woods 1984). As with the other data collection methods, there are several pros and cons to using archive data (Chappell 1994).

The first advantage of archive data is that the data are real in that they are typically provided by participants in the incident. The second is that there are large numbers of observations available (in contrast to accident data). The third is that the data have high ecological validity because it relates to real incidents that occurred under normal operating conditions. Finally, the cost of collecting the data is generally low, because the data already exist.

The main disadvantage of using archive data is that they usually have not been gathered for the specific purpose of the investigation at hand. The data therefore often have to be re-ordered and possibly re-represented before it can be appropriately analyzed. The second disadvantage is that the detail in the reports may not have been validated. The third is that the data may be subject to reporter biases, in terms of who reports, and the information that gets reported which may be biased by selective retrieval and rational reconstruction (Ericsson and Simon 1993).

10.2.4 Selecting the Most Appropriate Data Collection Method

The aim of each of the methods described above is to generate data that can be used to develop theories and models of human performance. Over the last 20 years, since the earliest work there has been a shift towards an increased use of real world data (e.g., Hutchins 1995), especially when studying expert performance in dynamic environments. The ideal method, however, would combine the contextual richness of real world situations with some of the experimental control of laboratory conditions. The work on the HEAP (Hollnagel et al. 1996) comes close to this ideal. The HEAP involved observing teams of operators running a high fidelity nuclear power plant simulator. Even with the HEAP work, however,

there were problems. The operators either knew or could guess the purpose of the study. As a result they had a higher than normal expectation that an abnormal situation would arise during a simulator session. The sessions were also usually limited to 1 h rather than the duration of a normal shift.

If you are designing an interface for a nuclear power plant control system, for example, you may regard ecological validity as being of paramount importance, and therefore decide against the use of laboratory experiments. Similarly, the hit and miss nature of using field observation for relatively infrequent events mitigates against its use, except perhaps in longitudinal studies or where unusual resources are available. Archive data, on the other hand, naturally have high ecological validity—a judicious choice of data source should make it possible to gain access to the hundreds of instances of human error required for analysis (Baxter 2000).

Ultimately your choice of method will depend on the weights you give to the pros and cons of the individual methods. In addition, however, you should make sure that you take into account the availability and access to the resources you will need to carry out your study.

10.3 Error Taxonomies

Most scientific study is underpinned by a well-defined classification system or taxonomy of the relevant phenomena. The taxonomy provides a frame of reference for the study, and enables other researchers to evaluate the results of that study. Although several human error taxonomies have been developed, there is no universal taxonomy that serves all the various purposes of error research (Senders and Moray 1991). Below we consider three well known examples.

The critical factor in generating a taxonomy of erroneous behavior is the choice of level of abstraction to use for categorization. Determining an appropriate level of abstraction requires that the purpose of the research be clearly defined. A useful rule of thumb is to model behavior at a level that allows remedies to be generated to facilitate the avoidance of a repetition of the same type of error in similar circumstances in the future.

10.3.1 The Technique for Human Error Rate Prediction

The taxonomy used in the Technique for Human Error Rate Prediction (THERP, Swain and Guttman 1983) is based around the commonly used notions of errors of omission and commission. In this taxonomy, actions can either be:

- Correct.
- Errors of omission: actions that are omitted. It may be difficult, however, to determine whether an action has been omitted or has just been delayed for a long time.

Table 10.1 GEMS taxonomy

Planning	Knowledge-based mistakes
	Rule-based mistakes
Storage	Skill-based lapses
Execution	Skill-based slips

- Errors of commission: actions that are performed inadequately, out of sequence, or at the wrong time (too early/late); or are qualitatively incorrect (too much/too little/wrong direction).
- Extraneous actions: actions that would not normally be expected at that particular time and place.

In THERP the probabilities of the different errors occurring are conditioned by performance shaping factors (PSFs). The PSFs are intended to take some account of the context in which the error occurs, to improve the accuracy of the individual error probability estimates. They are usually divided into external PSFs (such as situational characteristics), stressor PSFs (such as psychological stressors), and internal PSFs (such as organismic factors like previous training).

10.3.2 Generic Error Modeling System

Rasmussen's (1976) SRK model of behavior (see Chap. 5) has been used as the basis for several taxonomies that describe human performance in the operation of complex control systems, including Reason's (1990) Generic Error Modelling System (GEMS). Reason distinguishes between three types of errors: slips, lapses, and mistakes (see also Norman 1981). Slips (execution failures) occur when the person has the right intention but performs the wrong action. Lapses (memory storage failures) occur between the formulation of an intention and the execution of some action. Mistakes (intention failures) occur when the person initiates the wrong plan of action for the task at hand. Within the GEMS taxonomy, errors can be associated with planning, with storage, or with execution of actions (see Table 10.1).

Skill-based errors (slips and lapses) are normally attributable to monitoring failures. In particular, they are often linked to a lack of attention, whether deliberate or unintentional.

The rule- and knowledge-based errors (mistakes) are usually associated with problem solving activities. At the rule-based level, mistakes can arise when good rules are misapplied, or bad rules are applied. At the knowledge-based level, mistakes usually arise due to a lack of expertise, because people do not have all the knowledge required to perform the task at hand. Figure 10.2 shows the relationship between the different types of errors in GEMS and a typical stage model of human information processing.

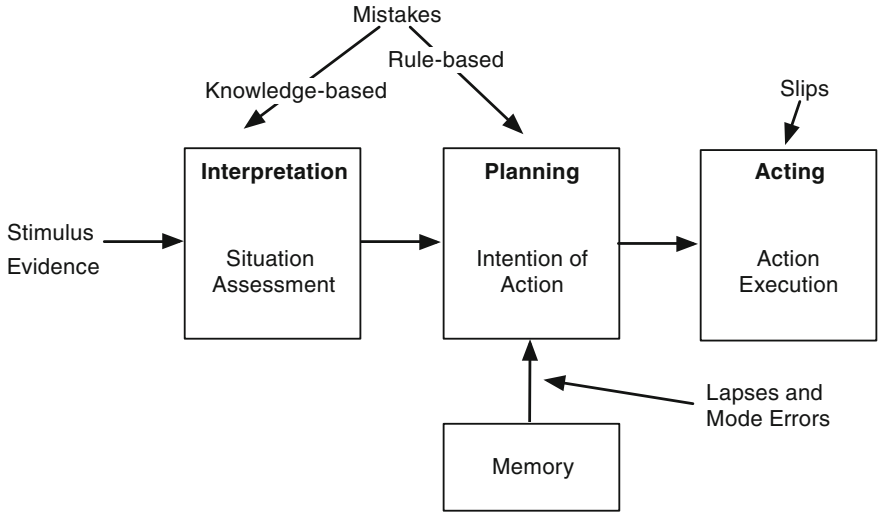

Fig. 10.2 Reason's Generic Error Modelling System (GEMS) error types related to a simple stage model of human information processing (adapted from Wickens and Holland 2000)

10.3.3 The Cognitive Reliability and Error Analysis Method

Hollnagel's (1998) Cognitive Reliability and Error Analysis Method (CREAM) tacitly acknowledges the fact that there is no single ideal taxonomy. Instead, the CREAM provides a generic framework for developing a domain specific taxonomy of causes and effects of erroneous actions.

In the CREAM, performance takes place in a context that is defined by the interaction between three high level factors:

1. The human operator
2. The technology (usually the system being operated by the human)
3. The wider organization (including the environment in which the system is located).

Each of these factors is explicitly accounted for in the CREAM that includes a scheme for classifying actions and events. The CREAM also takes account of the current consensus view that there is not a separate uniquely identifiable part of the human physiology that can be conveniently labeled *error generator*. As noted earlier, in many cases erroneous behavior occurs when the user takes what appears to be the appropriate action in conditions that are similar to, but not quite the same as what the operator believes them to be.

Any erroneous actions that can be detected can be categorized as belonging to one or more of eight possible error modes or effects—Hollnagel (1993b) also

Table 10.2 The CREAM genotypes of human error

Person-related	Observation
	Interpretation
	Planning
	Temporary
	Permanent
Organization-related	Communication
	Training
	Ambient conditions
	Working conditions
Technology-related	Equipment failure
	Procedures
	Temporary interface problems
	Permanent interface problems

refers to these as the logical phenotypes of human error—each of which can manifest itself in several ways (shown in italics):

- Timing: the action is *too early/too late/omitted*.
- Duration: the action is *too long/too short*.
- Force: the action uses *too much/too little force*.
- Distance: the action is carried on *too far/too short*.
- Speed: the action is *too fast/too slow*.
- Direction: the action is performed in the *wrong direction* or involves the *wrong type of movement*.
- Object: the action was carried out on a *proximal object/similar object/unrelated object* rather than the required object.
- Sequence: in a sequence of actions there was an *omission/skip forward/skip backward/repetition/reversal* or the *wrong action* was performed.

In addition, the CREAM makes provision for the inclusion of correct actions. Hollnagel (1998) suggests that a category labeled *no erroneous action* should be incorporated, either by adding it to each of the classification groups or by keeping it as a separate group.

The category of error mode to which a particular instance of an erroneous action belongs may not always be immediately obvious. In such cases, the particular situation at hand has to be carefully considered before any judgment can be made. This need for considered judgments is critical to the field of human error research, because the labeling of a particular action as being erroneous is invariably a judgment made in hindsight (Woods et al. 1994).

The other part of the CREAM taxonomy is made up of the possible causes, or genotypes. These are divided into three main categories as shown in Table 10.2. There is one category for each of the main factors that contribute to system performance: people, technology, and context (described as organization-related in the CREAM).

10.4 Analyzing Errors

There are many techniques available for analyzing errors, and any of them will usually provide some useful insights to help you understand what happened. Here we briefly discuss four techniques. Two have been widely used for several years in the safety systems engineering community. The others are more recent, and are less widely used, but offer interesting (and useful) perspectives for analyzing errors. Irrespective of which technique you choose (including those not covered here), you should make sure that it can be applied systematically.

10.4.1 Event Trees

Event trees are a bottom-up (inductive) technique for analyzing errors. They show the sequences of events that lead to all the possible outcomes. The trees are based on simple binary logic: at each node in the tree there are two possible branches based on whether an event does or does not happen (or whether a component failed or did not fail). The trees start with an initiating event, and are generated by thinking of all the possible consequences at each node in the tree. Each of the events can be assigned a probability (the sum of the probabilities for each of the two branches for a single node must add up to 1). The probability of all the identified outcomes can be calculated by multiplying together (ANDing) all the event probabilities along the path that leads from the initiating event to the outcome.

Figure 10.3 shows a quantified event tree for the case where a fire breaks out in an office block. The initiating event (shown at the left of the tree) is the fact that the fire starts, and the estimated frequency of this occurrence is one per year. The likelihood of the various resultant events (outcomes) is calculated by multiplying the appropriate probabilities together. The probability of multiple fatalities for this scenario, for example, is 0.015 (i.e., $0.1 \times 0.3 \times 0.5$).

10.4.2 Fault Trees

Fault trees are similar to event trees, although they are generated in a top down (deductive) manner, starting with the outcome and working backwards in time to try to find all the things that could have caused that particular outcome. Fault trees do not have to be binary trees, and an outcome can be determined by ANDing and ORing together a set of causal factors, as appropriate. Fault trees are normally only concerned with immediate effects, rather than the creation of latent conditions that can lie dormant within a system until some particular trigger activates them.

Fault trees can be either qualitative or quantified. To quantify a fault tree, a probability of occurrence is allocated to each of the lowest level leaf nodes in the

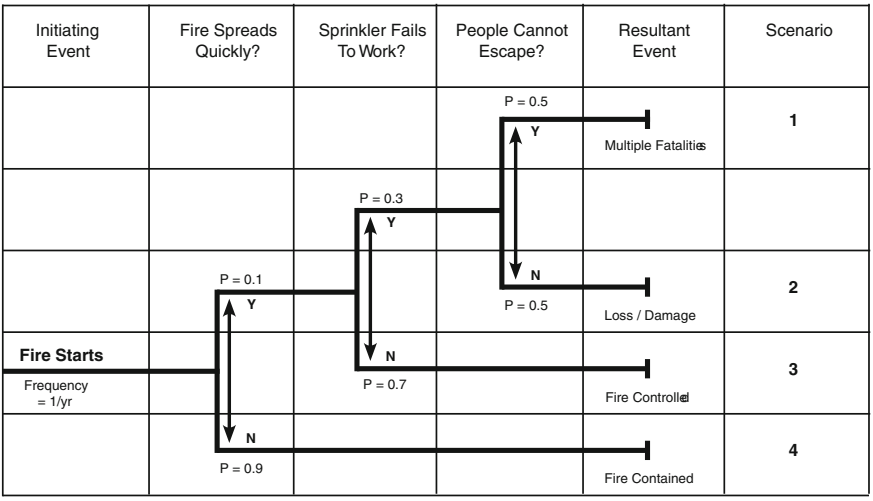

Fig. 10.3 Example quantification of event tree for a building protected by a sprinkler system (reproduced with permission from the Institution for Engineering and Technology)

tree. Fault trees can also be modified to include recovery actions; the resultant trees are usually called recovery trees (van der Schaaf 1991).

Figure 10.4 shows an example of a quantified fault tree for accidents at a particular road junction. The legend at the left of the figure describes the probabilities used in the figure. Cars driving too fast at this road junction, for example, occur frequently, so it is given a probability of 0.1.

10.4.3 CREAM

The CREAM can be used both for the retrospective analysis of accidents and the prospective analysis of possible errors in a system that is being designed. In both cases, the method that is followed is essentially the same.

Here we will only focus on the simple retrospective use of the CREAM (as shown in Fig. 10.5). The process starts with the description of the initiating event (which could be an accident or incident). This description needs to provide enough detail to form the basis for the analysis. From this description it should be possible to identify the error mode(s) (or phenotypes) associated with the event. The next step is to try and identify the possible antecedents for that error mode, using the set of tables of antecedents and consequents that lie at the heart of the CREAM method. If a specific antecedent is found, or there are no general antecedents for the error mode, then the analysis is complete. If a general antecedent is found then the next step is to find the general consequent associated with that antecedent. If none can be found, the analysis terminates, otherwise we use the general consequent as a specific consequent, and go through the loop again, this time trying to find a matching

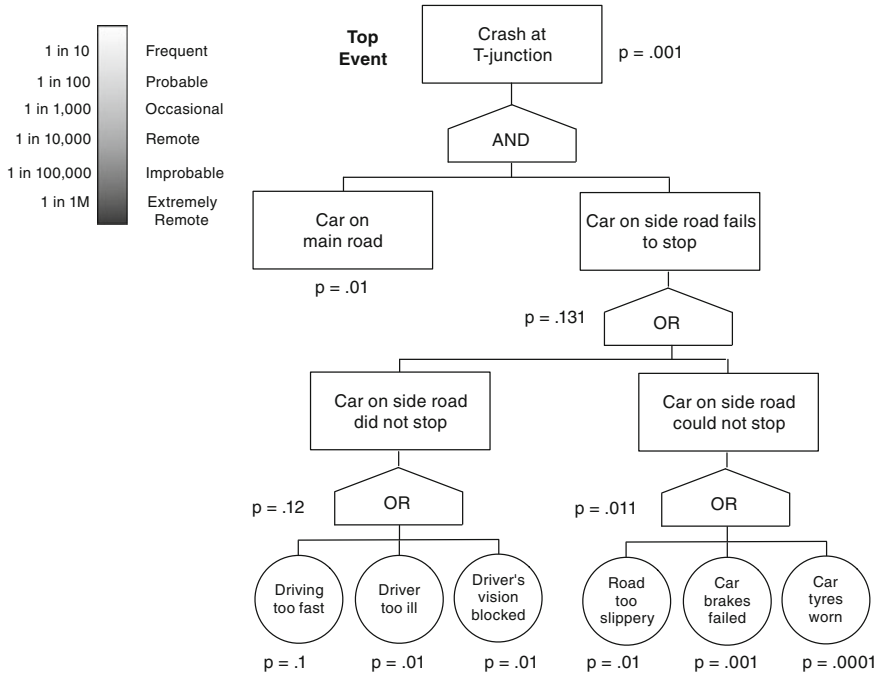

Fig. 10.4 A quantified fault tree showing the likelihood of a crash occurring at a given road junction (reproduced with permission from the Institution for Engineering and Technology)

antecedent for the specific consequent (rather than the error mode). As the process continues, a structure that looks like a tree of antecedents and consequents is built up. Note that these are really the *possible* antecedents and *possible* consequents. Some of them may be ruled out by what really happened during the event.

Using the CREAM to analyze errors can become quite involved, and is beyond the scope of this book. Hollnagel's (1998) book includes a good example of an analysis of a railway accident that occurred in New York in 1995 which is worth looking at, for those who are interested. Although the CREAM has not been widely adopted—partly because of a lack of tool support[1]—it does offer a nice illustration of how taking a different view of errors can generate new insights.

10.4.4 THEA

The Technique for Human Error Assessment (Pocock et al. 2001) assumes, like the CREAM, that the context in which actions are performed is one of the major influences on human performance. Like the CREAM, THEA is an iterative

[1] Although there is a browser based tool at http://www.ews.uiuc.edu/~serwy/cream/.

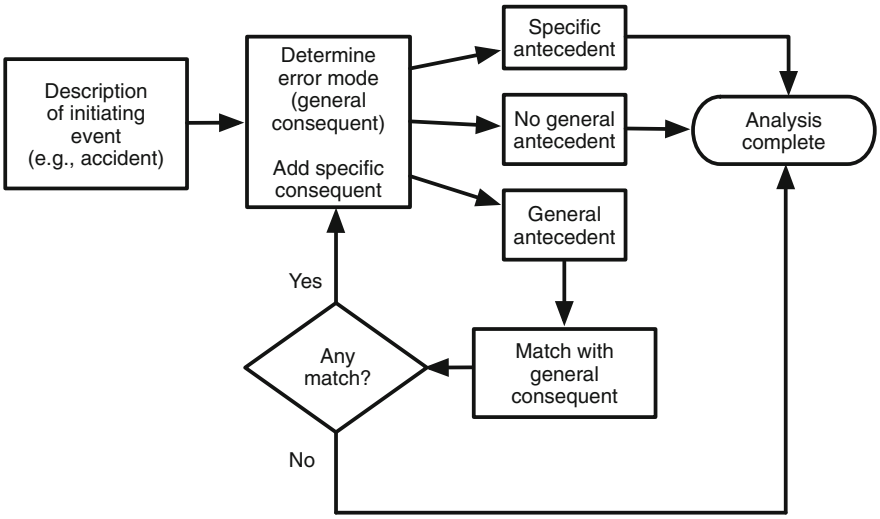

Fig. 10.5 CREAM. The basic CREAM analysis process (redrawn by the authors)

process, although the iterations take place at a higher level, i.e., the level of the design of the system or device.

The THEA process (see Fig. 10.6) starts with a description of the system in terms of its functional behavior, its interface, and how it communicates with other systems. This description is accompanied by a corresponding description of the work performed by the system. This comprises one (or more) descriptive scenarios, particularly focusing on the potential vulnerabilities of the system, and a description of the sorts of tasks that will be performed in terms of goals, actions, and plans.

The system description and work descriptions are used to structure the scenario. Any appropriate method that can be used to decompose goals, such as hierarchical task analysis (described in Chap. 11), can be used.

Once you have a structured scenario, you can do the error analysis. The analysis is based on a set of questions and a model of human information processing. Although the THEA reference guide uses Norman's (1988/2013) cyclic model of interaction, the process is model independent.

The final step is to consider the probability of the occurrence of the identified errors, and then make suggestions about changes to the design to deal with them. The process then iterates until some stopping point is determined, which will usually be down to the judgment of the designer.

As with the CREAM, THEA's uptake has been somewhat limited but, like the CREAM, it also shows how a systematic approach to analyzing errors can be used to inform design. Those who want more information should consult the THEA reference guide. THEA was originally designed to analyze situations involving a single person using a single artifact, whereas nowadays most work is performed by

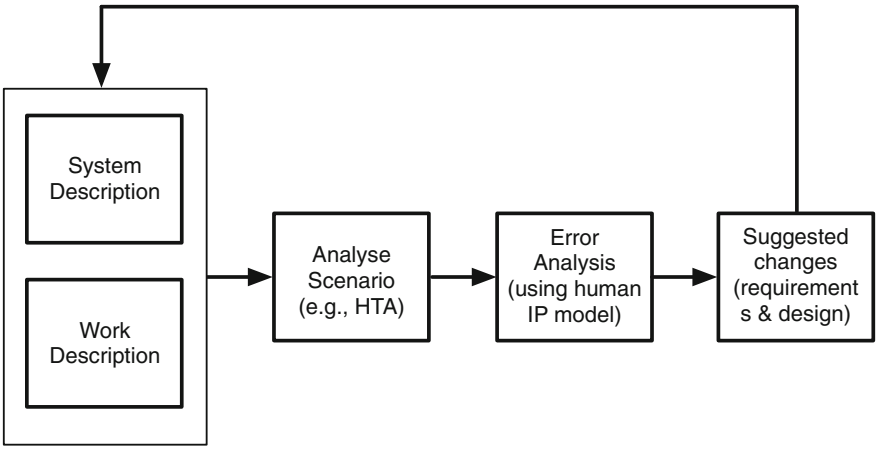

Fig. 10.6 The THEA process

teams of people. THEA was therefore adapted and extended to create a new method, CHLOE. CHLOE takes into account how people collaborate to get work done, and has been used to analyze errors in Air Traffic Control (Miguel and Wright 2003).

10.5 Implications for System Design

We need to consider how people interact with technology in a particular context when designing systems. Each of the components (people, technology, and context) can give rise to errors, so we need to design systems to take account of this, and try either to prevent the errors or at least to mitigate their consequences. It is also vital that we consider the interdependencies between people, technology, and context, because these can also give rise to errors.

We know that most systems can be described as socio-technical systems, in that they have social (people-related) components and technological components. Unfortunately, however, many designers think this means that the system can be decomposed into a social subsystem and a technical subsystem. In reality, such an atomistic decomposition is inappropriate, because of the interactions and interdependencies between the system's social and technical components. If these interactions and interdependencies are ignored, the emergent system behaviors (including errors) that they give rise to may be overlooked. Many system developers who claim that they use a socio-technical approach often decompose the system into social and technical subsystems, and focus most of their attention on the technical subsystem.

Allied to the decomposition into human and technical components is a technique called function allocation. When designing a system, you will need to

identify the list of functions that the system has to perform. These functions are then allocated to either the human or the machine using a static approach that is often based on Fitts' (1951) list, which is also referred to as the MABA-MABA (Men Are Better At–Machines Are Better At) approach. The problem with this approach is that designers often allocate all the tasks that they know how to automate to the technology, and then leave the human to carry out all the others. If we want to allocate functions effectively we need to consider the processing characteristics of both the humans and the technology so that we can reduce the chances of errors whilst performing a particular function.

If your system requires a lot of functions to be carried out in the same time frame, you may overload the operator (and the technology), thereby increasing the chances of an error occurring. In this case you may want to consider whether you can allocate functions dynamically, allowing tasks to be shed, and reallocated as workloads change. So, if operators get really busy, it should be possible for them to hand off tasks to the automation to reduce their workload, thereby improving overall system performance, and vice versa.

One of the ironies of automation (Bainbridge 1987) is that the more complex socio-technical systems become, the more we rely on people to intervene to fix them when errors occur. You will therefore often hear people talking about the need to keep people in the loop. It is important that the users are kept aware of what the system is doing, by providing them with feedback about the system's state. They can use this to detect errors, and to update their own mental model of how the system is working. It is also important that users are given the opportunity to practice their skills, so they do not forget how to carry out particular tasks, especially those that they perform infrequently. This is one of the reasons why aircraft pilots have to undergo recurrent training, and are expected to hand fly their aircraft on a fairly regular basis.

10.6 Summary

Human error is a complex subject, for several reasons:

- Confusion over the term itself, which gets used to describe the action, the cause of that action, and the consequence of that action too.
- The same factors govern the expression of both expertise and error.
- Some of the contributors are latent, and lie hidden, waiting for other triggering or potentiating factors.
- Errors are judgments made in hindsight and require some measure of expected human performance for comparison.
- Human performance involves a distributed system of people interacting with technology at the sharp end and organizational elements at the blunt end.
- Decisions made at the blunt end of the system can constrain the way that work is carried out at the sharp end.

- The way technology is deployed shapes human performance, creating the potential for new forms of error and failure.

Errors will happen. You may be able to get some idea of the sorts of errors that may occur with your system by looking at archive data, where it exists. Alternatively, you may be able to collect data by running experiments using your system or an appropriate simulator.

You can take appropriate account of potential errors by carefully considering the type of system that you are designing, the people who will use it, and the context in which they will operate it. There are several methods that will help you analyze your system for potential errors, including Event Trees, Fault Trees, the CREAM, and THEA, even at design time.

10.7 Other Resources

It is worth looking at Sidney Dekker's books to get a fuller understanding of the new (some would say more enlightened) view of human error. *Ten Questions About Human Error* (Dekker 2005) is an easy and entertaining read, whilst also being thought provoking. In it he addresses the issue of human error by posing the following questions, and then going on to explain and to answer them at length:

1. Was it mechanical failure or human error?
2. Why do safe systems fail?
3. Why are doctors more dangerous than gun owners?
4. Don't errors exist?
5. If you lose situation awareness, what replaces it?
6. Why do operators become complacent?
7. Why don't they follow procedures?
8. Can we automate error out of the system?
9. Will the system be safe?
10. Should we hold people accountable for their mistakes?

You should be able to start to answer at least some of these questions for yourself at this point.

The problems of dealing with blame, and how to establish a just (i.e., fair) culture, form the content of Dekker's book, *Just Culture* (Dekker 2007). In it he gives several examples of the sorts of problems that can occur when trying to make sure that justice (in the widest sense of the word, rather than just the legalistic view) is served. The main focus of the book is on trying to balance safety and accountability so that people who make honest mistakes are not necessarily held to be culpable. He dispels the simplistic idea about needing to punish those that cross the line, by showing that who gets to draw the line, and where they get to draw it, are major determinants in deciding whether someone will be regarded as culpable or not.

10.8 Exercises

10.1 Redraw Fig. 10.3, the event tree, for errors when entering a purchase on a smartphone or other mobile device. You can make (and may need to make) assumptions about the application and about people. Note these assumptions, and note briefly what studies or references you would need to read to find out more accurate answers.

10.2 Interact through several transactions with an online commerce site, such as abebooks.com, or an online library, or other online service that delivers information. Keep a log of errors you make using a keystroke logger, an observer, or video. Analyze these errors for frequency and type using two different taxonomies.

10.3 What are the error rates while typing? When you read a finished document, it looks like there are none. In this exercise, gather some data on error rates while typing. You can do this in several ways. You could ask people to type without looking at what they are typing. This would give an uncorrected error rate. You could show them a paper to type, and check how many letters are different between the source and their typing (using the Unix 'diff' tool, or using Word's compare documents). You could also set up a web page and see how many times a user clicks on the correct link when asked. In your analyses you should consider what errors are more common, and what may lead to increased errors.

10.4 In 2009 a man drove his $1 million Bugatti Veyron car into a lake. He blamed it on dropping his cell phone. Extend the fault tree in Fig. 10.4 to include the effect of cell phones on accidents. To do this, you will have to note where cell phones will interact with driving, and you will have to attempt to provide quantitative measures of how often events happen with a cell phone.

References

Air Accidents Investigation Branch. (1989). Report on the accident to Boeing 737-400- G-OBME near Kegworth, Leicestershire. Retrieved 8 March 2014, from http://www.aaib.gov.uk/publications/formal_reports/4_1990_g_obme.cfm

Arnstein, F. (1997). Catalogue of human error. *British Journal of Anaesthesia, 79*, 645–656.

Bainbridge, L. (1987). Ironies of automation. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error* (pp. 271–283). Chicester: John Wiley.

Baxter, G. D. (2000). *State misinterpretation in flight crew behaviour: An incident-based analysis.* Unpublished PhD Thesis, University of Nottingham.

Besnard, D., Greathead, D., & Baxter, G. (2004). When mental models go wrong. Co-occurrences in dynamic, critical systems. *International Journal of Human-Computer Studies, 60*(60), 117–128.

Bogner, M. S. (Ed.). (2004). *Misadventures in health care*. Mahwah, NJ: Erlbaum.

Chappell, S. L. (1994). Using voluntary incident reports for human factors evaluations. In N. Johnston, N. McDonald, & R. Fuller (Eds.), *Aviation psychology in practice* (pp. 149–169). Aldershot, UK: Avebury.

Dekker, S. (2005). *Ten questions about human error: A new view of human factors and system safety*. Mahwah, NJ: Erlbaum.
Dekker, S. (2007). *Just culture: Balancing safety and accountability*. Aldershot, Hampshire, England: Ashgate Publishing.
Dismukes, K., Young, G., & Sumwalt, R. (1998). Cockpit interruptions and distractions: Effective management requires a careful balancing act. *ASRS Directline, 10*, 4–9.
Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* (2nd ed.). Cambridge, MA: MIT Press.
Fitts, P. M. (1951). *Human engineering for an effective air navigation and traffic control system*. Washington, DC: National Research Council.
Gigerenzer, G. (2004). Dread risk, September 11, and fatal traffic accidents. *Psychological Science, 15*(4), 286–287.
Hollnagel, E. (1993a). *Human reliability analysis: Context and control*. London: Academic Press.
Hollnagel, E. (1993b). The phenotypes of erroneous actions. *International Journal of Man-Machine Studies, 39*, 1–32.
Hollnagel, E. (1998). *Cognitive reliability and error assessment method*. Oxford, UK: Elsevier Science.
Hollnagel, E., Drøivoldsmo, A., & Kirwan, B. (1996). Practical insights from studies of operator diagnosis. In *Proceedings of ECCE-8. Eighth European Conference on Cognitive Ergonomics* (pp. 133–137). Granada, 8–12 Sept, 1996. Rocquencourt, France: European Association of Cognitive Ergonomics.
Hollnagel, E., Woods, D. D., & Leveson, N. (Eds.). (2006). *Resilience engineering: Concepts and precepts*. Aldershot, UK: Ashton Press.
Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.
Johnson, C. W., & Holloway, C. M. (2007). A longitudinal analysis of the causal factors in major maritime accidents in the USA and Canada (1996–2006). In *The Safety of Systems: Proceedings of the 15th Safety-Critical Systems Symposium* (pp. 85–104). London, UK: Springer.
Kemeny (chairman), J. G. (1979). *The need for change: The Legacy of TMI*. Washington, DC: The President's Commission on the accident at TMI.
Mach, E. (1905). *Knowledge and error* (English Trans., D. Reidel, 1976). Dordrecht, Netherlands: Reidel.
Miguel, A., & Wright, P. (2003). CHLOE: A technique for analysing collaborative systems. In *Proceedings of 9th Conference on Cognitive Science Approaches to Process Control* (pp. 53–60). New York, NY: ACM Press.
Nardi, B. A. (1996). *Context and consciousness: Activity theory and human-computer interaction*. Cambridge, MA: MIT Press.
Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 88*, 1–15.
Norman, D. A. (1988). *The psychology of everyday things*. New York, NY: Basic Books.
Norman, D. A. (2013). *The design of everyday things*. New York, NY: Basic Books.
Petroski, H. (1985/1992). *To engineer is human: The role of failure in successful design*. New York, NY: Vintage Books.
Petroski, H. (1994). *Design paradigms: Case histories of error and judgment in engineering*. Cambridge, UK: Cambridge University Press.
Petroski, H. (2006). *Success through failure: The paradox of design*. Princeton, NJ: Princeton University Press.
Pew, R. W., Miller, D. C., & Feeher, C. E. (1981). *Evaluation of proposed control room improvements through analysis of critical operator decisions (EPRI-NP-1982)*. Cambridge, MA: Bolt, Beranek & Newman.
Pocock, S., Harrison, M., Wright, P., & Johnson, P. (2001). THEA: A technique for human error assessment early in design. In *Proceedings of Human-Computer Interaction: INTERACT'01* (pp. 247–254). Amsterdam, The Netherlands: IOS Press.
Randell, B. (2000). Facing up to faults. *The Computer Journal, 43*, 95–106.

Rasmussen, J. (1976). Outlines of a hybrid model of the process operator. In T. G. Sheridan & G. Johannsen (Eds.), *Monitoring behavior and supervisory control* (pp. 371–383). New York, NY: Plenum.

Rasmussen, J. (1980). What can be learned from human error reports? In K. Duncan, M. Gruneberg, & D. Wallis (Eds.), *Changes in working life* (pp. 97–113). Chichester, UK: Wiley.

Rasmussen, J. (1988). Human error mechanisms in complex work environments. *Reliability Engineering and System Safety, 22*, 155–167.

Rasmussen, J., Pejtersen, A.-M., & Goodstein, L. P. (1994). *Cognitive systems engineering*. Chichester, UK: Wiley.

Reason, J. (1990). *Human error*. Cambridge, UK: Cambridge University Press.

Senders, J. W., & Moray, N. P. (1991). *Human error: Cause, prediction, and reduction*. Hillsdale, NJ: Erlbaum.

Swain, A. D., & Guttman, H. E. (1983). *A handbook of human reliability analysis with emphasis on nuclear power applications*. Washington, DC: US Nuclear Regulatory Commission.

van der Schaaf, T. W. (1991). A framework for designing near miss management systems. In T. W. v. d. Schaaf, D. A. Lucas & A. Hale (Eds.), *Near miss reporting as a safety tool* (pp. 27–35). Oxford, UK: Butterworth-Heinemann.

Vaughan, D. (1997). *The Challenger launch decision*. Chicago, IL: University of Chicago Press.

Wagenaar, W. A., & Groeneweg, J. (1987). Accidents at sea: Multiple causes and impossible consequences. *International Journal of Man-Machine Studies, 27*, 587–598.

Wickens, C. D., & Hollands, J. G. (2000). Engineering psychology and human performance (3rd ed.). Upper Saddle River, NJ: Prentice-Hall.

Wiener, E., Kanki, B., & Helmreich, R. L. (Eds.). (1993). *Cockpit resource management*. London, UK: Academic Press.

Woods, D. D. (1984). Some results on operator performance in emergency events. *Institution of Chemical Engineers Symposium Series* (Vol. 90, pp. 21–31).

Woods, D. D., Johannesen, L. J., Cook, R. I., & Sarter, N. B. (1994). *Behind human error: Cognitive systems, computers, and hindsight*. Wright-Patterson Air Force Base, OH: CSERIAC.