4 A Systems-Theoretic View of Causality

In the traditional causality models, accidents are considered to be caused by chains of failure events, each failure directly causing the next one in the chain. Part I explained why these simple models are no longer adequate for the more complex sociotechnical systems we are attempting to build today. The definition of accident causation needs to be expanded beyond failure events so that it includes component interaction accidents and indirect or systemic causal mechanisms.

The first step is to generalize the definition of an accident.[1] An *accident* is an unplanned and undesired loss event. That loss may involve human death and injury, but it may also involve other major losses, including mission, equipment, financial, and information losses.

Losses result from component failures, disturbances external to the system, interactions among system components, and behavior of individual system components that lead to hazardous system states. Examples of hazards include the release of toxic chemicals from an oil refinery, a patient receiving a lethal dose of medicine, two aircraft violating minimum separation requirements, and commuter train doors opening between stations.[2]

In systems theory, emergent properties, such as safety, arise from the interactions among the system components. The emergent properties are controlled by imposing constraints on the behavior of and interactions among the components. Safety then becomes a *control* problem where the goal of the control is to enforce the safety constraints. Accidents result from inadequate control or enforcement of safety-related constraints on the development, design, and operation of the system.

At Bhopal, the safety constraint that was violated was that the MIC must not come in contact with water. In the Mars Polar Lander, the safety constraint was that the spacecraft must not impact the planet surface with more than a maximum force.

1. A set of definitions used in this book can be found in appendix A.

2. Hazards are more carefully defined in chapter 7.

In the batch chemical reactor accident described in chapter 2, one safety constraint is a limitation on the temperature of the contents of the reactor.

The problem then becomes one of control where the goal is to control the behavior of the system by enforcing the safety constraints in its design and operation. Controls must be established to accomplish this goal. These controls need not necessarily involve a human or automated controller. Component behavior (including failures) and unsafe interactions may be controlled through physical design, through process (such as manufacturing processes and procedures, maintenance processes, and operations), or through social controls. Social controls include organizational (management), governmental, and regulatory structures, but they may also be cultural, policy, or individual (such as self-interest). As an example of the latter, one explanation that has been given for the 2009 financial crisis is that when investment banks went public, individual controls to reduce personal risk and long-term profits were eliminated and risk shifted to shareholders and others who had few and weak controls over those taking the risks.

In this framework, understanding why an accident occurred requires determining why the control was ineffective. Preventing future accidents requires shifting from a focus on preventing failures to the broader goal of designing and implementing controls that will enforce the necessary constraints.

The STAMP (System-Theoretic Accident Model and Processes) accident model is based on these principles. Three basic constructs underlie STAMP: safety constraints, hierarchical safety control structures, and process models.

4.1 Safety Constraints

The most basic concept in STAMP is not an event, but a constraint. Events leading to losses occur only because safety constraints were not successfully enforced.

The difficulty in identifying and enforcing safety constraints in design and operations has increased from the past. In many of our older and less automated systems, physical and operational constraints were often imposed by the limitations of technology and of the operational environments. Physical laws and the limits of our materials imposed natural constraints on the complexity of physical designs and allowed the use of passive controls.

In engineering, *passive controls* are those that maintain safety by their presence—basically, the system fails into a safe state or simple interlocks are used to limit the interactions among system components to safe ones. Some examples of passive controls that maintain safety by their presence are shields or barriers such as containment vessels, safety harnesses, hardhats, passive restraint systems in vehicles, and fences. Passive controls may also rely on physical principles, such as gravity, to fail into a safe state. An example is an old railway semaphore that used weights

to ensure that if the cable (controlling the semaphore) broke, the arm would automatically drop into the STOP position. Other examples include mechanical relays designed to fail with their contacts open, and retractable landing gear for aircraft in which the wheels drop and lock in the landing position if the pressure system that raises and lowers them fails. For the batch chemical reactor example in chapter 2, where the order valves are opened is crucial, designers might have used a physical interlock that did not allow the catalyst valve to be opened while the water valve was closed.

In contrast, *active controls* require some action(s) to provide protection: (1) *detection* of a hazardous event or condition (monitoring), (2) *measurement* of some variable(s), (3) interpretation of the measurement (*diagnosis*), and (4) *response* (recovery or fail-safe procedures), all of which must be completed before a loss occurs. These actions are usually implemented by a control system, which now commonly includes a computer.

Consider the simple passive safety control where the circuit for a high-power outlet is run through a door that shields the power outlet. When the door is opened, the circuit is broken and the power disabled. When the door is closed and the power enabled, humans cannot touch the high power outlet. Such a design is simple and foolproof. An active safety control design for the same high power source, requires some type of sensor to detect when the access door to the power outlet is opened and an active controller to issue a control command to cut the power. The failure modes for the active control system are greatly increased over the passive design, as is the complexity of the system component interactions. In the railway semaphore example, there must be a way to detect that the cable has broken (probably now a digital system is used instead of a cable so the failure of the digital signaling system must be detected) and some type of active controls used to warn operators to stop the train. The design of the batch chemical reactor described in chapter 2 used a computer to control the valve opening and closing order instead of a simple mechanical interlock.

While simple examples are used here for practical reasons, the complexity of our designs is reaching and exceeding the limits of our intellectual manageability with a resulting increase in component interaction accidents and lack of enforcement of the system safety constraints. Even the relatively simple computer-based batch chemical reactor valve control design resulted in a component interaction accident. There are often very good reasons to use active controls instead of passive ones, including increased functionality, more flexibility in design, ability to operate over large distances, weight reduction, and so on. But the difficulty of the engineering problem is increased and more potential for design error is introduced.

A similar argument can be made for the interactions between operators and the processes they control. Cook [40] suggests that when controls were primarily

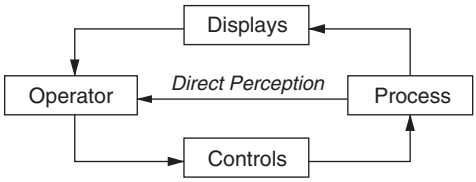


Figure 4.1
Operator has direct perception of process and mechanical controls.

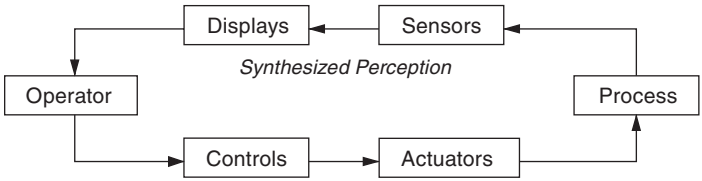


Figure 4.2
Operator has indirect information about process state and indirect controls.

mechanical and were operated by people located close to the operating process, proximity allowed sensory perception of the status of the process via direct physical feedback such as vibration, sound, and temperature (figure 4.1). Displays were directly linked to the process and were essentially a physical extension of it. For example, the flicker of a gauge needle in the cab of a train indicated that (1) the engine valves were opening and closing in response to slight pressure fluctuations, (2) the gauge was connected to the engine, (3) the pointing indicator was free, and so on. In this way, the displays provided a rich source of information about the controlled process and the state of the displays themselves.

The introduction of electromechanical controls allowed operators to control processes from a greater distance (both physical and conceptual) than possible with pure mechanically linked controls (figure 4.2). That distance, however, meant that operators lost a lot of direct information about the process—they could no longer sense the process state directly and the control and display surfaces no longer provided as rich a source of information about the process or the state of the controls themselves. The system designers had to synthesize and provide an image of the process state to the operators. An important new source of design errors was introduced by the need for the designers to determine beforehand what information the operator would need under all conditions to safely control the process. If the designers had not anticipated a particular situation could occur and provided for it in the original system design, they might also not anticipate the need of the operators for information about it during operations.

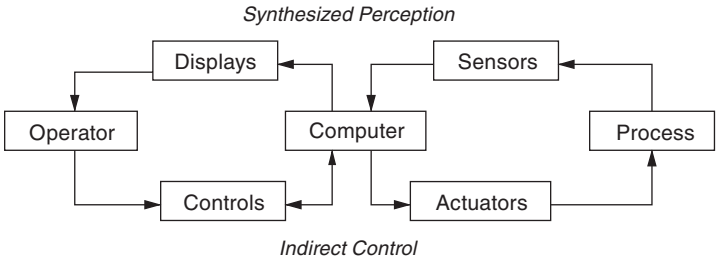


Figure 4.3
Operator has computer-generated displays and controls the process through a computer.

Designers also had to provide feedback on the actions of the operators and on any failures that might have occurred. The controls could now be operated without the desired effect on the process, and the operators might not know about it. Accidents started to occur due to incorrect feedback. For example, major accidents (including Three Mile Island) have involved the operators commanding a valve to open and receiving feedback that the valve had opened, when in reality it had not. In this case and others, the valves were wired to provide feedback indicating that power had been applied to the valve, but not that the valve had actually opened. Not only could the design of the feedback about success and failures of control actions be misleading in these systems, but the return links were also subject to failure.

Electromechanical controls relaxed constraints on the system design allowing greater functionality (figure 4.3). At the same time, they created new possibilities for designer and operator error that had not existed or were much less likely in mechanically controlled systems. The later introduction of computer and digital controls afforded additional advantages and removed even more constraints on the control system design—and introduced more possibility for error. Proximity in our old mechanical systems provided rich sources of feedback that involved almost all of the senses, enabling early detection of potential problems. We are finding it hard to capture and provide these same qualities in new systems that use automated controls and displays.

It is the freedom from constraints that makes the design of such systems so difficult. Physical constraints enforced discipline and limited complexity in system design, construction, and modification. The physical constraints also shaped system design in ways that efficiently transmitted valuable physical component and process information to operators and supported their cognitive processes.

The same argument applies to the increasing complexity in organizational and social controls and in the interactions among the components of sociotechnical systems. Some engineering projects today employ thousands of engineers. The Joint

Strike Fighter, for example, has eight thousand engineers spread over most of the United States. Corporate operations have become global, with greatly increased interdependencies and producing a large variety of products. A new holistic approach to safety, based on control and enforcing safety constraints in the entire sociotechnical system, is needed to ensure safety.

To accomplish this goal, system-level constraints must be identified, and responsibility for enforcing them must be divided up and allocated to appropriate groups. For example, the members of one group might be responsible for performing hazard analyses. The manager of this group might be assigned responsibility for ensuring that the group has the resources, skills, and authority to perform such analyses and for ensuring that high-quality analyses result. Higher levels of management might have responsibility for budgets, for establishing corporate safety policies, and for providing oversight to ensure that safety policies and activities are being carried out successfully and that the information provided by the hazard analyses is used in design and operations.

During system and product design and development, the safety constraints will be broken down and sub-requirements or constraints allocated to the components of the design as it evolves. In the batch chemical reactor, for example, the system safety requirement is that the temperature in the reactor must always remain below a particular level. A design decision may be made to control this temperature using a reflux condenser. This decision leads to a new constraint: "Water must be flowing into the reflux condenser whenever catalyst is added to the reactor." After a decision is made about what component(s) will be responsible for operating the catalyst and water valves, additional requirements will be generated. If, for example, a decision is made to use software rather than (or in addition to) a physical interlock, the software must be assigned the responsibility for enforcing the constraint: "The water valve must always be open when the catalyst valve is open."

In order to provide the level of safety demanded by society today, we first need to identify the safety constraints to enforce and then to design effective controls to enforce them. This process is much more difficult for today's complex and often high-tech systems than in the past and new techniques, such as those described in part III, are going to be required to solve it, for example, methods to assist in generating the component safety constraints from the system safety constraints. The alternative—building only the simple electromechanical systems of the past or living with higher levels of risk—is for the most part not going to be considered an acceptable solution.

4.2 The Hierarchical Safety Control Structure

In systems theory (see section 3.3), systems are viewed as hierarchical structures, where each level imposes constraints on the activity of the level beneath it—that is,

constraints or lack of constraints at a higher level allow or control lower-level behavior.

Control processes operate between levels to control the processes at lower levels in the hierarchy. These control processes enforce the safety constraints for which the control process is responsible. Accidents occur when these processes provide inadequate control and the safety constraints are violated in the behavior of the lower-level components.

By describing accidents in terms of a hierarchy of control based on adaptive feedback mechanisms, adaptation plays a central role in the understanding and prevention of accidents.

At each level of the hierarchical structure, inadequate control may result from missing constraints (unassigned responsibility for safety), inadequate safety control commands, commands that were not executed correctly at a lower level, or inadequately communicated or processed feedback about constraint enforcement. For example, an operations manager may provide unsafe work instructions or procedures to the operators, or the manager may provide instructions that enforce the safety constraints, but the operators may ignore them. The operations manager may not have the feedback channels established to determine that unsafe instructions were provided or that his or her safety-related instructions are not being followed.

Figure 4.4 shows a typical sociotechnical hierarchical safety control structure common in a regulated, safety-critical industry in the United States, such as air transportation. Each system, of course, must be modeled to include its specific features. Figure 4.4 has two basic hierarchical control structures—one for system development (on the left) and one for system operation (on the right)—with interactions between them. An aircraft manufacturer, for example, might have only system development under its immediate control, but safety involves both development and operational use of the aircraft, and neither can be accomplished successfully in isolation: Safety during operation depends partly on the original design and development and partly on effective control over operations. Communication channels may be needed between the two structures.[3] For example, aircraft manufacturers must communicate to their customers the assumptions about the operational environment upon which the safety analysis was based, as well as information about safe operating procedures. The operational environment (e.g., the commercial airline industry), in turn, provides feedback to the manufacturer about the performance of the system over its lifetime.

Between the hierarchical levels of each safety control structure, effective communication channels are needed, both a downward *reference channel* providing the

3. Not all interactions between the two control structures are shown in the figure to simplify it and make it more readable.

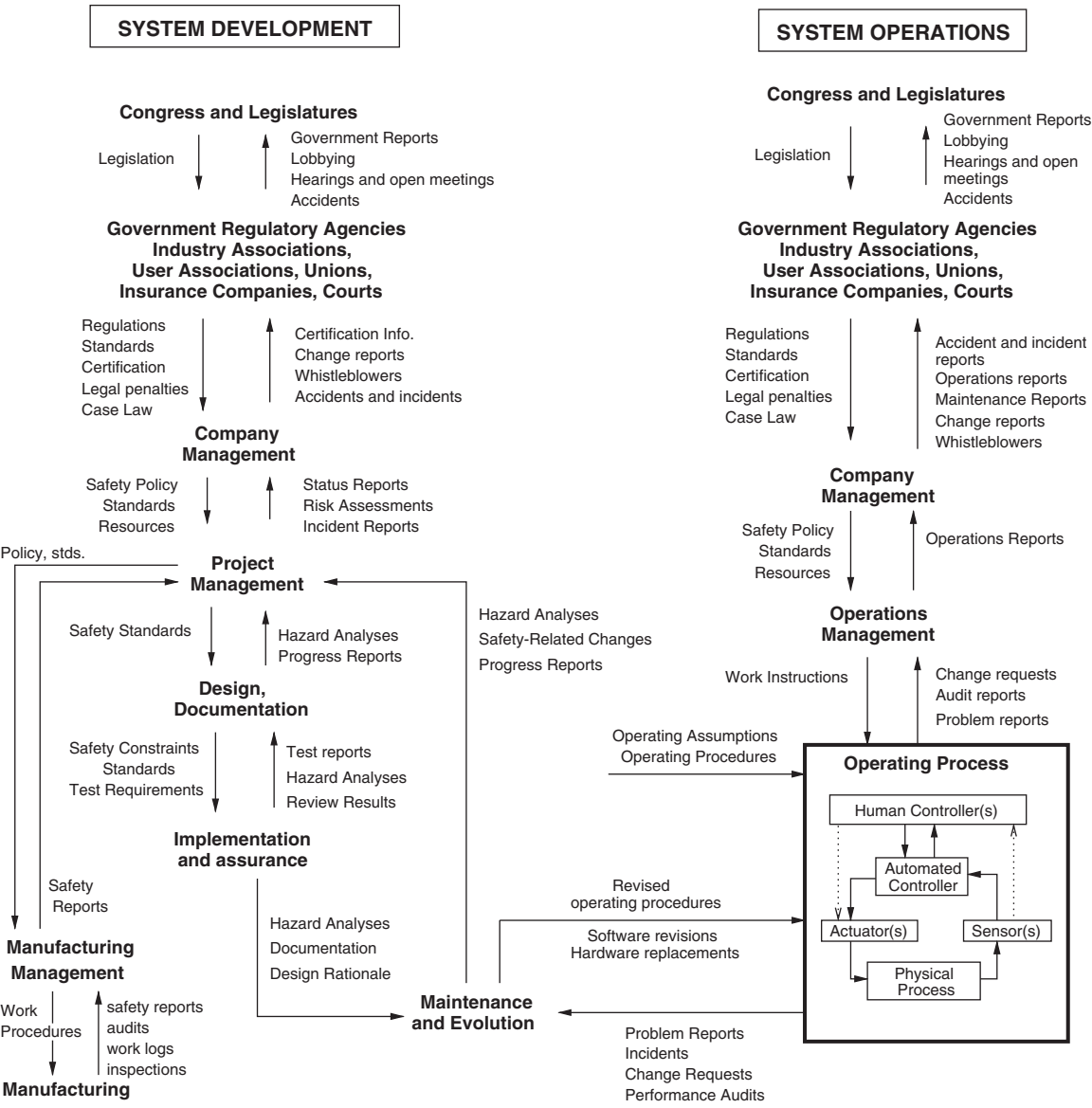


Figure 4.4
General form of a model of sociotechnical control.

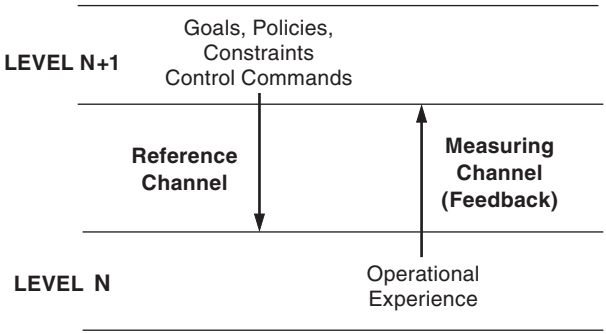


Figure 4.5
Communication channels between control levels.

information necessary to impose safety constraints on the level below and an upward *measuring channel* to provide feedback about how effectively the constraints are being satisfied (figure 4.5). Feedback is critical in any open system in order to provide adaptive control. The controller uses the feedback to adapt future control commands to more readily achieve its goals.

Government, general industry groups, and the court system are the top two levels of each of the generic control structures shown in figure 4.4. The government control structure in place to control development may differ from that controlling operations—responsibility for certifying the aircraft developed by aircraft manufacturers is assigned to one group at the FAA, while responsibility for supervising airline operations is assigned to a different group. The appropriate constraints in each control structure and at each level will vary but in general may include technical design and process constraints, management constraints, manufacturing constraints, and operational constraints.

At the highest level in both the system development and system operation hierarchies are Congress and state legislatures.[4] Congress controls safety by passing laws and by establishing and funding government regulatory structures. Feedback as to the success of these controls or the need for additional ones comes in the form of government reports, congressional hearings and testimony, lobbying by various interest groups, and, of course, accidents.

The next level contains government regulatory agencies, industry associations, user associations, insurance companies, and the court system. Unions have always played an important role in ensuring safe operations, such as the air traffic controllers union in the air transportation system, or in ensuring worker safety in

4. Obvious changes are required in the model for countries other than the United States. The United States is used in the example because of the author's familiarity with it.

manufacturing. The legal system tends to be used when there is no regulatory authority and the public has no other means to encourage a desired level of concern for safety in company management. The constraints generated at this level and imposed on companies are usually in the form of policy, regulations, certification, standards (by trade or user associations), or threat of litigation. Where there is a union, safety-related constraints on operations or manufacturing may result from union demands and collective bargaining.

Company management takes the standards, regulations, and other general controls on its behavior and translates them into specific policy and standards for the company. Many companies have a general safety policy (it is required by law in Great Britain) as well as more detailed standards documents. Feedback may come in the form of status reports, risk assessments, and incident reports.

In the development control structure (shown on the left of figure 4.4), company policies and standards are usually tailored and perhaps augmented by each engineering project to fit the needs of the particular project. The higher-level control process may provide only general goals and constraints and the lower levels may then add many details to operationalize the general goals and constraints given the immediate conditions and local goals. For example, while government or company standards may require a hazard analysis be performed, the system designers and documenters (including those designing the operational procedures and writing user manuals) may have control over the actual hazard analysis process used to identify specific safety constraints on the design and operation of the system. These detailed procedures may need to be approved by the level above.

The design constraints identified as necessary to control system hazards are passed to the implementers and assurers of the individual system components along with standards and other requirements. Success is determined through feedback provided by test reports, reviews, and various additional hazard analyses. At the end of the development process, the results of the hazard analyses as well as documentation of the safety-related design features and design rationale should be passed on to the maintenance group to be used in the system evolution and sustainment process.

A similar process involving layers of control is found in the system operation control structure. In addition, there will be (or at least should be) interactions between the two structures. For example, the safety design constraints used during development should form the basis for operating procedures and for performance and process auditing.

As in any control loop, time lags may affect the flow of control actions and feedback and may impact the effectiveness of the control loop in enforcing the safety constraints. For example, standards can take years to develop or change—a time scale that may keep them behind current technology and practice. At the physical

level, new technology may be introduced in different parts of the system at different rates, which may result in *asynchronous evolution* of the control structure. In the accidental shootdown of two U.S. Army Black Hawk helicopters by two U.S. Air Force F-15s in the no-fly zone over northern Iraq in 1994, for example, the fighter jet aircraft and the helicopters were inhibited in communicating by radio because the F-15 pilots used newer jam-resistant radios that could not communicate with the older-technology Army helicopter radios. Hazard analysis needs to include the influence of these time lags and potential changes over time.

A common way to deal with time lags leading to delays is to delegate responsibility to lower levels that are not subject to as great a delay in obtaining information or feedback from the measuring channels. In periods of quickly changing technology, time lags may make it necessary for the lower levels to augment the control processes passed down from above or to modify them to fit the current situation. Time lags at the lowest levels, as in the Black Hawk shootdown example, may require the use of feedforward control to overcome lack of feedback or may require temporary controls on behavior: Communication between the F-15s and the Black Hawks would have been possible if the F-15 pilots had been told to use an older radio technology available to them, as they were commanded to do for other types of friendly aircraft.

More generally, control structures always change over time, particularly those that include humans and organizational components. Physical devices also change with time, but usually much slower and in more predictable ways. If we are to handle social and human aspects of safety, then our accident causality models must include the concept of change. In addition, controls and assurance that the safety control structure remains effective in enforcing the constraints over time are required.

Control does not necessarily imply rigidity and authoritarian management styles. Rasmussen notes that control at each level may be enforced in a very prescriptive command and control structure or it may be loosely implemented as performance objectives with many degrees of freedom in how the objectives are met [165]. Recent trends from management by *oversight* to management by *insight* reflect differing levels of feedback control that are exerted over the lower levels and a change from prescriptive management control to management by objectives, where the objectives are interpreted and satisfied according to the local context.

Management insight, however, does not mean abdication of safety-related responsibility. In a Milstar satellite loss [151] and both the Mars Climate Orbiter [191] and Mars Polar Lander [95, 213] losses, the accident reports all note that a poor transition from oversight to insight was a factor in the losses. Attempts to delegate decisions and to manage by objectives require an explicit formulation of the value criteria to be used and an effective means for communicating the values down through society and organizations. In addition, the impact of specific decisions at

each level on the objectives and values passed down need to be adequately and formally evaluated. Feedback is required to measure how successfully the functions are being performed.

Although regulatory agencies are included in the figure 4.4 example, there is no implication that government regulation is required for safety. The only requirement is that responsibility for safety is distributed in an appropriate way throughout the sociotechnical system. In aircraft safety, for example, manufacturers play the major role while the FAA type certification authority simply provides oversight that safety is being successfully engineered into aircraft at the lower levels of the hierarchy. If companies or industries are unwilling or incapable of performing their public safety responsibilities, then government has to step in to achieve the overall public safety goals. But a much better solution is for company management to take responsibility, as it has direct control over the system design and manufacturing and over operations.

The safety-control structure will differ among industries and examples are spread among the following chapters. Figure C.1 in appendix C shows the control structure and safety constraints for the hierarchical water safety control system in Ontario, Canada. The structure is drawn on its side (as is more common for control diagrams) so that the top of the hierarchy is on the left side of the figure. The system hazard is exposure of the public to *E. coli* or other health-related contaminants through the public drinking water system; therefore, the goal of the safety control structure is to prevent such exposure. This goal leads to two system safety constraints:

1. Water quality must not be compromised.
2. Public health measures must reduce the risk of exposure if water quality is somehow compromised (such as notification and procedures to follow).

The physical processes being controlled by this control structure (shown at the right of the figure) are the water system, the wells used by the local public utilities, and public health. Details of the control structure are discussed in appendix C, but appropriate responsibility, authority, and accountability must be assigned to each component with respect to the role it plays in the overall control structure. For example, the responsibility of the Canadian federal government is to establish a nationwide public health system and ensure that it is operating effectively. The provincial government must establish regulatory bodies and codes, provide resources to the regulatory bodies, provide oversight and feedback loops to ensure that the regulators are doing their job adequately, and ensure that adequate risk assessment is conducted and effective risk management plans are in place. Local public utility operations must apply adequate doses of chlorine to kill bacteria, measure the chlorine residuals, and take further steps if evidence of bacterial contamination is

found. While chlorine residuals are a quick way to get feedback about possible contamination, more accurate feedback is provided by analyzing water samples but takes longer (it has a greater time lag). Both have their uses in the overall safety control structure of the public water supply.

Safety control structures may be very complex: Abstracting and concentrating on parts of the overall structure may be useful in understanding and communicating about the controls. In examining different hazards, only subsets of the overall structure may be relevant and need to be considered in detail and the rest can be treated as the inputs to or the environment of the substructure. The only critical part is that the hazards must first be identified at the system level and the process must then proceed top-down and not bottom-up to identify the safety constraints for the parts of the overall control structure.

The operation of sociotechnical safety control structures at all levels is facing the stresses noted in chapter 1, such as rapidly changing technology, competitive and time-to-market pressures, and changing public and regulatory views of responsibility for safety. These pressures can lead to a need for new procedures or new controls to ensure that required safety constraints are not ignored.

4.3 Process Models

The third concept used in STAMP, along with safety constraints and hierarchical safety control structures, is process models. Process models are an important part of control theory. The four conditions required to control a process are described in chapter 3. The first is a *goal*, which in STAMP is the safety constraints that must be enforced by each controller in the hierarchical safety control structure. The *action condition* is implemented in the (downward) control channels and the *observability condition* is embodied in the (upward) feedback or measuring channels. The final condition is the *model condition*: *Any* controller—human or automated—needs a model of the process being controlled to control it effectively (figure 4.6).

At one extreme, this process model may contain only one or two variables, such as the model required for a simple thermostat, which contains the current temperature and the setpoint and perhaps a few control laws about how temperature is changed. At the other extreme, effective control may require a very complex model with a large number of state variables and transitions, such as the model needed to control air traffic.

Whether the model is embedded in the control logic of an automated controller or in the mental model maintained by a human controller, it must contain the same type of information: the required relationship among the system variables (the control laws), the current state (the current values of the system variables), and the ways the process can change state. This model is used to determine what control

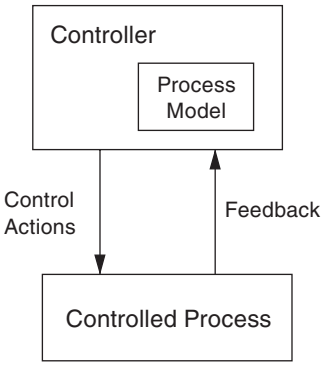


Figure 4.6
Every controller must contain a model of the process being controlled. Accidents can occur when the controller's process model does not match the system being controlled and the controller issues unsafe commands.

actions are needed, and it is updated through various forms of feedback. If the model of the room temperature shows that the ambient temperature is less than the setpoint, then the thermostat issues a control command to start a heating element. Temperature sensors provide feedback about the (hopefully rising) temperature. This feedback is used to update the thermostat's model of the current room temperature. When the setpoint is reached, the thermostat turns off the heating element. In the same way, human operators also require accurate process or mental models to provide safe control actions.

Component interaction accidents can usually be explained in terms of incorrect process models. For example, the Mars Polar Lander software thought the spacecraft had landed and issued a control instruction to shut down the descent engines. The captain of the *Herald of Free Enterprise* thought the ferry doors were closed and ordered the ship to leave the mooring. The pilots in the Cali Colombia B757 crash thought *R* was the symbol denoting the radio beacon near Cali.

In general, accidents often occur, particularly component interaction accidents and accidents involving complex digital technology or human error, when the process model used by the controller (automated or human) does not match the process and, as a result:

1. Incorrect or unsafe control commands are given
2. Required control actions (for safety) are not provided
3. Potentially correct control commands are provided at the wrong time (too early or too late), or
4. Control is stopped too soon or applied too long.

These four types of inadequate control actions are used in the new hazard analysis technique described in chapter 8.

A model of the process being controlled is required not just at the lower physical levels of the hierarchical control structure, but at all levels. In order to make proper decisions, the manager of an oil refinery may need to have a model of the current maintenance level of the safety equipment of the refinery, the state of safety training of the workforce, and the degree to which safety requirements are being followed or are effective, among other things. The CEO of the global oil conglomerate has a much less detailed model of the state of the refineries he controls but at the same time requires a broader view of the state of safety of all the corporate assets in order to make appropriate corporate-level decisions impacting safety.

Process models are not only used during operations but also during system development activities. Designers use both models of the system being designed and models of the development process itself. The developers may have an incorrect model of the system or software behavior necessary for safety or the physical laws controlling the system. Safety may also be impacted by developers' incorrect models of the development process itself.

As an example of the latter, a Titan/Centaur satellite launch system, along with the Milstar satellite it was transporting into orbit, was lost due to a typo in a load tape used by the computer to determine the attitude change instructions to issue to the engines. The information on the load tape was essentially part of the process model used by the attitude control software. The typo was not caught during the development process partly because of flaws in the developers' models of the testing process—each thought someone else was testing the software using the actual load tape when, in fact, nobody was (see appendix B).

In summary, process models play an important role (1) in understanding why accidents occur and why humans provide inadequate control over safety-critical systems and (2) in designing safer systems.

4.4 STAMP

The STAMP (Systems-Theoretic Accident Model and Process) model of accident causation is built on these three basic concepts—safety constraints, a hierarchical safety control structure, and process models—along with basic systems theory concepts. All the pieces for a new causation model have been presented. It is now simply a matter of putting them together.

In STAMP, systems are viewed as interrelated components kept in a state of dynamic equilibrium by feedback control loops. Systems are not treated as static but as dynamic processes that are continually adapting to achieve their ends and to react to changes in themselves and their environment.

Safety is an emergent property of the system that is achieved when appropriate constraints on the behavior of the system and its components are satisfied. The original design of the system must not only enforce appropriate constraints on behavior to ensure safe operation, but the system must continue to enforce the safety constraints as changes and adaptations to the system design occur over time.

Accidents are the result of flawed processes involving interactions among people, societal and organizational structures, engineering activities, and physical system components that lead to violating the system safety constraints. The process leading up to an accident is described in STAMP in terms of an adaptive feedback function that fails to maintain safety as system performance changes over time to meet a complex set of goals and values.

Instead of defining safety management in terms of preventing component failures, it is defined as creating a safety control structure that will enforce the behavioral safety constraints and ensure its continued effectiveness as changes and adaptations occur over time. Effective safety (and risk) management may require limiting the types of changes that occur but the goal is to allow as much flexibility and performance enhancement as possible while enforcing the safety constraints.

Accidents can be understood, using STAMP, by identifying the safety constraints that were violated and determining why the controls were inadequate in enforcing them. For example, understanding the Bhopal accident requires determining not simply why the maintenance personnel did not insert the slip blind, but also why the controls that had been designed into the system to prevent the release of hazardous chemicals and to mitigate the consequences of such occurrences—including maintenance procedures and oversight of maintenance processes, refrigeration units, gauges and other monitoring units, a vent scrubber, water spouts, a flare tower, safety audits, alarms and practice alerts, emergency procedures and equipment, and others—were not successful.

STAMP not only allows consideration of more accident causes than simple component failures, but it also allows more sophisticated analysis of failures and component failure accidents. Component failures may result from inadequate constraints on the manufacturing process; inadequate engineering design such as missing or incorrectly implemented fault tolerance; lack of correspondence between individual component capacity (including human capacity) and task requirements; unhandled environmental disturbances (e.g., electromagnetic interference or EMI); inadequate maintenance; physical degradation (wearout); and so on.

Component failures may be prevented by increasing the integrity or resistance of the component to internal or external influences or by building in safety margins or safety factors. They may also be avoided by operational controls, such as

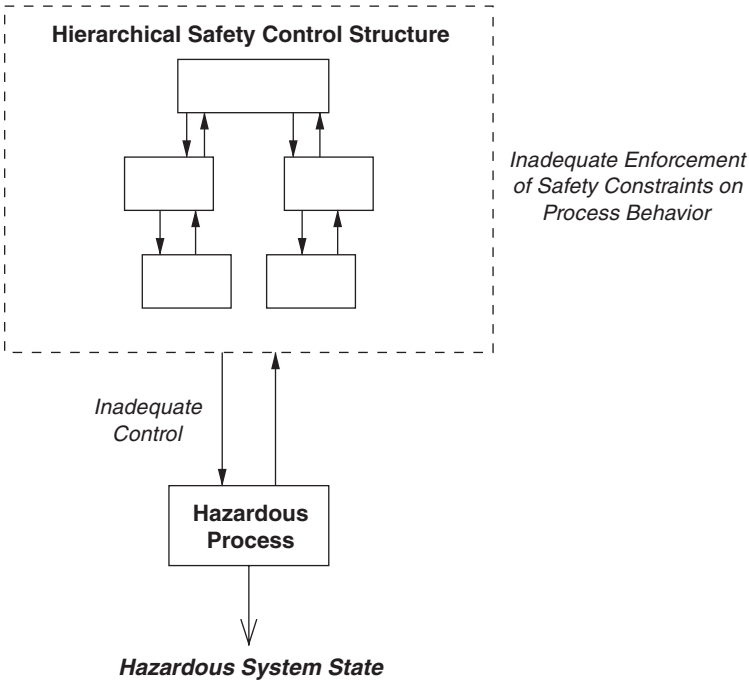


Figure 4.7
Accidents result from inadequate enforcement of the behavioral safety constraints on the process.

operating the component within its design envelope and by periodic inspections and preventive maintenance. Manufacturing controls can reduce deficiencies or flaws introduced during the manufacturing process. The effects of physical component failure on system behavior may be eliminated or reduced by using redundancy. The important difference from other causality models is that STAMP goes beyond simply blaming component failure for accidents by requiring that the reasons be identified for why those failures occurred (including systemic factors) and led to an accident, that is, why the controls instituted for preventing such failures or for minimizing their impact on safety were missing or inadequate. And it includes other types of accident causes, such as component interaction accidents, which are becoming more frequent with the introduction of new technology and new roles for humans in system control.

STAMP does not lend itself to a simple graphic representation of accident causality (see figure 4.7). While dominoes, event chains, and holes in Swiss cheese are very compelling because they are easy to grasp, they oversimplify causality and thus the approaches used to prevent accidents.

4.5 A General Classification of Accident Causes

Starting from the basic definitions in STAMP, the general causes of accidents can be identified using basic systems and control theory. The resulting classification is useful in accident analysis and accident prevention activities.

Accidents in STAMP are the result of a complex process that results in the system behavior violating the safety constraints. The safety constraints are enforced by the control loops between the various levels of the hierarchical control structure that are in place during design, development, manufacturing, and operations.

Using the STAMP causality model, if there is an accident, one or more of the following must have occurred:

1. The safety constraints were not enforced by the controller.
   a. The control actions necessary to enforce the associated safety constraint at each level of the sociotechnical control structure for the system were not provided.
   b. The necessary control actions were provided but at the wrong time (too early or too late) or stopped too soon.
   c. Unsafe control actions were provided that caused a violation of the safety constraints.
2. Appropriate control actions were provided but not followed.

These same general factors apply at each level of the sociotechnical control structure, but the interpretation (application) of the factor at each level may differ.

Classification of accident causal factors starts by examining each of the basic components of a control loop (see figure 3.2) and determining how their improper operation may contribute to the general types of inadequate control.

Figure 4.8 shows the classification. The causal factors in accidents can be divided into three general categories: (1) the controller operation, (2) the behavior of actuators and controlled processes, and (3) communication and coordination among controllers and decision makers. When humans are involved in the control structure, context and behavior-shaping mechanisms also play an important role in causality.

4.5.1 Controller Operation

Controller operation has three primary parts: control inputs and other relevant external information sources, the control algorithms, and the process model. Inadequate, ineffective, or missing control actions necessary to enforce the safety constraints and ensure safety can stem from flaws in each of these parts. For human controllers and actuators, context is also an important factor.

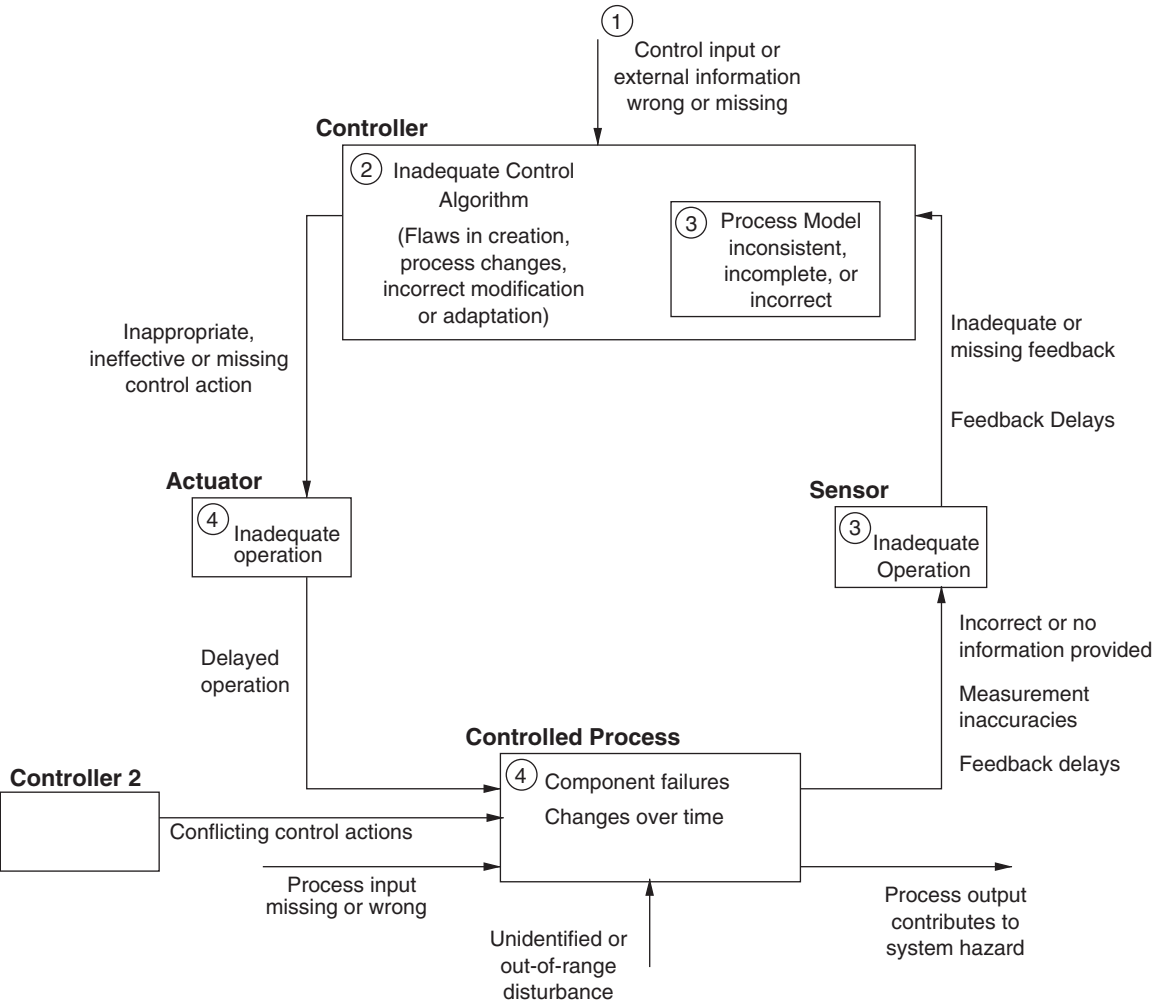


Figure 4.8
A classification of control flaws leading to hazards.

Unsafe Inputs (① in figure 4.8)

Each controller in the hierarchical control structure is itself controlled by higher-level controllers. The control actions and other information provided by the higher level and required for safe behavior may be missing or wrong. Using the Black Hawk friendly fire example again, the F-15 pilots patrolling the no-fly zone were given instructions to switch to a non-jammed radio mode for a list of aircraft types that did not have the ability to interpret jammed broadcasts. Black Hawk helicopters had not been upgraded with new anti-jamming technology but were omitted from the list and so could not hear the F-15 radio broadcasts. Other types of missing or wrong noncontrol inputs may also affect the operation of the controller.

Unsafe Control Algorithms (② in figure 4.8)

Algorithms in this sense are both the procedures designed by engineers for hardware controllers and the procedures that human controllers use. Control algorithms may not enforce safety constraints because the algorithms are inadequately designed originally, the process may change and the algorithms become unsafe, or the control algorithms may be inadequately modified by maintainers if the algorithms are automated or through various types of natural adaptation if they are implemented by humans. Human control algorithms are affected by initial training, by the procedures provided to the operators to follow, and by feedback and experimentation over time (see figure 2.9).

Time delays are an important consideration in designing control algorithms. Any control loop includes time lags, such as the time between the measurement of process parameters and receiving those measurements or between issuing a command and the time the process state actually changes. For example, pilot response delays are important time lags that must be considered in designing the control function for TCAS[5] or other aircraft systems, as are time lags in the controlled process—the aircraft trajectory, for example—caused by aircraft performance limitations.

Delays may not be directly observable, but may need to be inferred. Depending on where in the feedback loop the delay occurs, different control algorithms are required to cope with the delays [25]: dead time and time constants require an algorithm that makes it possible to predict when an action is needed before the need. Feedback delays generate requirements to predict when a prior control action has taken effect and when resources will be available again. Such requirements may impose the need for some type of open loop or feedforward strategy to cope with

5. TCAS (Traffic alert and Collision Avoidance System) is an airborne system used to avoid collisions between aircraft. More details about TCAS can be found in chapter 10.

delays. When time delays are not adequately considered in the control algorithm, accidents can result.

Leplat has noted that many accidents relate to *asynchronous evolution* [112], where one part of a system (in this case the hierarchical safety control structure) changes without the related necessary changes in other parts. Changes to subsystems may be carefully designed, but consideration of their effects on other parts of the system, including the safety control aspects, may be neglected or inadequate. Asynchronous evolution may also occur when one part of a properly designed system deteriorates.

In both these cases, the erroneous expectations of users or system components about the behavior of the changed or degraded subsystem may lead to accidents. The Ariane 5 trajectory changed from that of the Ariane 4, but the inertial reference system software was not changed. As a result, an assumption of the inertial reference software was violated and the spacecraft was lost shortly after launch. One factor in the loss of contact with SOHO (SOlar Heliospheric Observatory), a scientific spacecraft, in 1998 was the failure to communicate to operators that a functional change had been made in a procedure to perform gyro spin down. The Black Hawk friendly fire accident (analyzed in chapter 5) had several examples of asynchronous evolution, for example the mission changed and an individual key to communication between the Air Force and Army left, leaving the safety control structure without an important component.

Communication is a critical factor here as well as monitoring for changes that may occur and feeding back this information to the higher-level control. For example, the safety analysis process that generates constraints always involves some basic assumptions about the operating environment of the process. When the environment changes such that those assumptions are no longer true, as in the Ariane 5 and SOHO examples, the controls in place may become inadequate. Embedded pacemakers provide another example. These devices were originally assumed to be used only in adults, who would lie quietly in the doctor's office while the pacemaker was being "programmed." Later these devices began to be used in children, and the assumptions under which the hazard analysis was conducted and the controls were designed no longer held and needed to be revisited. A requirement for effective updating of the control algorithms is that the assumptions of the original (and subsequent) analysis are recorded and retrievable.

Inconsistent, Incomplete, or Incorrect Process Models (③ in figure 4.8)

Section 4.3 stated that effective control is based on a model of the process state. Accidents, particularly component interaction accidents, most often result from inconsistencies between the models of the process used by the controllers (both

human and automated) and the actual process state. When the controller's model of the process (either the human mental model or the software or hardware model) diverges from the process state, erroneous control commands (based on the incorrect model) can lead to an accident: for example, (1) the software does not know that the plane is on the ground and raises the landing gear, or (2) the controller (automated or human) does not identify an object as friendly and shoots a missile at it, or (3) the pilot thinks the aircraft controls are in *speed* mode but the computer has changed the mode to *open descent* and the pilot behaves inappropriately for that mode, or (4) the computer does not think the aircraft has landed and overrides the pilots' attempts to operate the braking system. All of these examples have actually occurred.

The mental models of the system developers are also important. During software development, for example, the programmers' models of required behavior may not match the engineers' models (commonly referred to as a software requirements error), or the software may be executed on computer hardware or may control physical systems during operations that differ from what was assumed by the programmer and used during testing. The situation becomes more even complicated when there are multiple controllers (both human and automated) because each of their process models must also be kept consistent.

The most common form of inconsistency occurs when one or more process models is incomplete in terms of not defining appropriate behavior for all possible process states or all possible disturbances, including unhandled or incorrectly handled component failures. Of course, no models are complete in the absolute sense: The goal is to make them complete enough that no safety constraints are violated when they are used. Criteria for completeness in this sense are presented in *Safeware*, and completeness analysis is integrated into the new hazard analysis method as described in chapter 9.

How does the process model become inconsistent with the actual process state? The process model designed into the system (or provided by training if the controller is human) may be wrong from the beginning, there may be missing or incorrect feedback for updating the process model as the controlled process changes state, the process model may be updated incorrectly (an error in the algorithm of the controller), or time lags may not be accounted for. The result can be uncontrolled disturbances, unhandled process states, inadvertent commanding of the system into a hazardous state, unhandled or incorrectly handled controlled process component failures, and so forth.

Feedback is critically important to the safe operation of the controller. A basic principle of system theory is that no control system will perform better than its measuring channel. Feedback may be missing or inadequate because such feedback is not included in the system design, flaws exist in the monitoring or feedback

communication channel, the feedback is not timely, or the measuring instrument operates inadequately.

A contributing factor cited in the Cali B757 accident report, for example, was the omission of the waypoints[6] behind the aircraft from cockpit displays, which contributed to the crew not realizing that the waypoint for which they were searching was behind them (missing feedback). The model of the Ariane 501 attitude used by the attitude control software became inconsistent with the launcher attitude when an error message sent by the inertial reference system was interpreted by the attitude control system as data (incorrect processing of feedback), causing the spacecraft onboard computer to issue an incorrect and unsafe command to the booster and main engine nozzles.

Other reasons for the process models to diverge from the true system state may be more subtle. Information about the process state has to be inferred from measurements. For example, in the TCAS II aircraft collision avoidance system, relative range positions of other aircraft are computed based on round-trip message propagation time. The theoretical control function (control law) uses the true values of the controlled variables or component states (e.g., true aircraft positions). However, at any time, the controller has only measured values, which may be subject to time lags or inaccuracies. The controller must use these measured values to infer the true conditions in the process and, if necessary, to derive corrective actions to maintain the required process state. In the TCAS example, sensors include on-board devices such as altimeters that provide measured altitude (not necessarily true altitude) and antennas for communicating with other aircraft. The primary TCAS actuator is the pilot, who may or may not respond to system advisories. The mapping between the measured or assumed values and the true values can be flawed.

To summarize, process models can be incorrect from the beginning—where correct is defined in terms of consistency with the current process state and with the models being used by other controllers—or they can become incorrect due to erroneous or missing feedback or measurement inaccuracies. They may also be incorrect only for short periods of time due to time lags in the process loop.

4.5.2 Actuators and Controlled Processes (④ in figure 4.8)

The factors discussed so far have involved inadequate control. The other case occurs when the control commands maintain the safety constraints, but the controlled process may not implement these commands. One reason might be a failure or flaw in the reference channel, that is, in the transmission of control commands. Another reason might be an actuator or controlled component fault or failure. A third is that

6. A *waypoint* is a set of coordinates that identify a point in physical space.

the safety of the controlled process may depend on inputs from other system components, such as power, for the execution of the control actions provided. If these process inputs are missing or inadequate in some way, the controller process may be unable to execute the control commands and accidents may result. Finally, there may be external disturbances that are not handled by the controller.

In a hierarchical control structure, the actuators and controlled process may themselves be a controller of a lower-level process. In this case, the flaws in executing the control are the same described earlier for a controller.

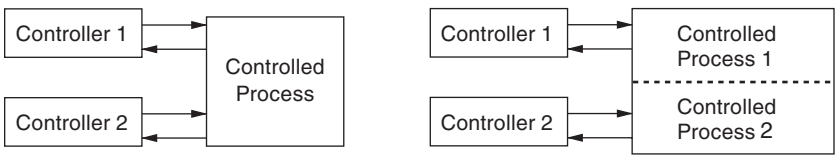
Once again, these types of flaws do not simply apply to operations or to the technical system but also to system design and development. For example, a common flaw in system development is that the safety information gathered or created by the system safety engineers (the hazards and the necessary design constraints to control them) is inadequately communicated to the system designers and testers, or that flaws exist in the use of this information in the system development process.

4.5.3 Coordination and Communication among Controllers and Decision Makers

When there are multiple controllers (human and/or automated), control actions may be inadequately coordinated, including unexpected side effects of decisions or actions or conflicting control actions. Communication flaws play an important role here.

Leplat suggests that accidents are most likely in *overlap areas* or in *boundary areas* or where two or more controllers (human or automated) control the same process or processes with common boundaries (figure 4.9) [112]. In both boundary and overlap areas, the potential exists for ambiguity and for conflicts among independent decisions.

Responsibility for the control functions in boundary areas is often poorly defined. For example, Leplat cites an iron and steel plant where frequent accidents occurred at the boundary of the blast furnace department and the transport department. One conflict arose when a signal informing transport workers of the state of the blast



(a) Example of an overlap

(b) Example of a boundary area

Figure 4.9
Problems often occur when there is shared control over the same process or at the boundary areas between separately controlled processes.

furnace did not work and was not repaired because each department was waiting for the other to fix it. Faverge suggests that such dysfunction can be related to the number of management levels separating the workers in the departments from a common manager: The greater the distance, the more difficult the communication, and thus the greater the uncertainty and risk.

Coordination problems in the control of boundary areas are rife. As mentioned earlier, a Milstar satellite was lost due to inadequate attitude control of the Titan/Centaur launch vehicle, which used an incorrect process model based on erroneous inputs on a software load tape. After the accident, it was discovered that nobody had tested the software using the actual load tape—each group involved in testing and assurance had assumed some other group was doing so. In the system development process, system engineering and mission assurance activities were missing or ineffective, and a common control or management function was quite distant from the individual development and assurance groups (see appendix B). One factor in the loss of the Black Hawk helicopters to friendly fire over northern Iraq was that the helicopters normally flew only in the boundary areas of the no-fly zone and procedures for handling aircraft in those areas were ill defined. Another factor was that an Army base controlled the flights of the Black Hawks, while an Air Force base controlled all the other components of the airspace. A common control point once again was high above where the accident occurred in the control structure. In addition, communication problems existed between the Army and Air Force bases at the intermediate control levels.

Overlap areas exist when a function is achieved by the cooperation of two controllers or when two controllers exert influence on the same object. Such overlap creates the potential for conflicting control actions (dysfunctional interactions among control actions). Leplat cites a study of the steel industry that found 67 percent of technical incidents with material damage occurred in areas of co-activity, although these represented only a small percentage of the total activity areas. In an A320 accident in Bangalore, India, the pilot had disconnected his flight director during approach and assumed that the copilot would do the same. The result would have been a mode configuration in which airspeed is automatically controlled by the autothrottle (the *speed* mode), which is the recommended procedure for the approach phase. However, the copilot had not turned off his flight director, which meant that *open descent* mode became active when a lower altitude was selected instead of *speed* mode, eventually contributing to the crash of the aircraft short of the runway [181]. In the Black Hawks' shootdown by friendly fire, the aircraft surveillance officer (ASO) thought she was responsible only for identifying and tracking aircraft south of the 36th Parallel, while the air traffic controller for the area north of the 36th Parallel thought the ASO was also tracking and identifying aircraft in his area and acted accordingly.

In 2002, two aircraft collided over southern Germany. An important factor in the accident was the lack of coordination between the airborne TCAS (collision avoidance) system and the ground air traffic controller. They each gave different and conflicting advisories on how to avoid a collision. If both pilots had followed one or the other, the loss would have been avoided, but one followed the TCAS advisory and the other followed the ground air traffic control advisory.

4.5.4 Context and Environment

Flawed human decision making can result from incorrect information and inaccurate process models, as described earlier. But human behavior is also greatly impacted by the context and environment in which the human is working. These factors have been called "behavior shaping mechanisms." While value systems and other influences on decision making can be considered to be inputs to the controller, describing them in this way oversimplifies their role and origin. A classification of the contextual and behavior-shaping mechanisms is premature at this point, but relevant principles and heuristics are elucidated throughout the rest of the book.

4.6 Applying the New Model

To summarize, STAMP focuses particular attention on the role of constraints in safety management. Accidents are seen as resulting from inadequate control or enforcement of constraints on safety-related behavior at each level of the system development and system operations control structures. Accidents can be understood in terms of why the controls that were in place did not prevent or detect maladaptive changes.

Accident causal analysis based on STAMP starts with identifying the safety constraints that were violated and then determines why the controls designed to enforce the safety constraints were inadequate or, if they were potentially adequate, why the system was unable to exert appropriate control over their enforcement.

In this conception of safety, there is no "root cause." Instead, the accident "cause" consists of an inadequate safety control structure that under some circumstances leads to the violation of a behavioral safety constraint. Preventing future accidents requires reengineering or designing the safety control structure to be more effective.

Because the safety control structure and the behavior of the individuals in it, like any physical or social system, changes over time, accidents must be viewed as dynamic processes. Looking only at the time of the proximal loss events distorts and omits from view the most important aspects of the larger accident process that are needed to prevent reoccurrences of losses from the same causes in the future. Without that view, we see and fix only the symptoms, that is, the results of the flawed processes and inadequate safety control structure without getting to the sources of those symptoms.

To understand the dynamic aspects of accidents, the process leading to the loss can be viewed as an adaptive feedback function where the safety control system performance degrades over time as the system attempts to meet a complex set of goals and values. Adaptation is critical in understanding accidents, and the adaptive feedback mechanism inherent in the model allows a STAMP analysis to incorporate adaptation as a fundamental system property.

We have found in practice that using this model helps us to separate factual data from the interpretations of that data: While the events and physical data involved in accidents may be clear, their importance and the explanations for why the factors were present are often subjective as is the selection of the events to consider.

STAMP models are also more complete than most accident reports and other models, for example see [9, 89, 140]. Each of the explanations for the incorrect FMS input of *R* in the Cali American Airlines accident described in chapter 2, for example, appears in the STAMP analysis of that accident at the appropriate levels of the control structure where they operated. The use of STAMP helps not only to identify the factors but also to understand the relationships among them.

While STAMP models will probably not be useful in law suits as they do not assign blame for the accident to a specific person or group, they do provide more help in understanding accidents by forcing examination of each part of the socio-technical system to see how it contributed to the loss—and there will usually be contributions at each level. Such understanding should help in learning how to engineer safer systems, including the technical, managerial, organizational, and regulatory aspects.

To accomplish this goal, a framework for classifying the factors that lead to accidents was derived from the basic underlying conceptual accident model (see figure 4.8). This classification can be used in identifying the factors involved in a particular accident and in understanding their role in the process leading to the loss. The accident investigation after the Black Hawk shootdown (analyzed in detail in the next chapter) identified 130 different factors involved in the accident. In the end, only the AWACS senior director was court-martialed, and he was acquitted. The more one knows about an accident process, the more difficult it is to find one person or part of the system responsible, but the easier it is to find effective ways to prevent similar occurrences in the future.

STAMP is useful not only in analyzing accidents that have occurred but in developing new and potentially more effective system engineering methodologies to prevent accidents. Hazard analysis can be thought of as investigating an accident before it occurs. Traditional hazard analysis techniques, such as fault tree analysis and various types of failure analysis techniques, do not work well for very complex systems, for software errors, human errors, and system design errors. Nor do they usually include organizational and management flaws. The problem is that these

hazard analysis techniques are limited by a focus on failure events and the role of component failures in accidents; they do not account for component interaction accidents, the complex roles that software and humans are assuming in high-tech systems, the organizational factors in accidents, and the indirect relationships between events and actions required to understand why accidents occur.

STAMP provides a direction to take in creating these new hazard analysis and prevention techniques. Because in a system accident model everything starts from constraints, the new approach focuses on identifying the constraints required to maintain safety; identifying the flaws in the control structure that can lead to an accident (inadequate enforcement of the safety constraints); and then designing a control structure, physical system and operating conditions that enforces the constraints.

Such hazard analysis techniques augment the typical failure-based design focus and encourage a wider variety of risk reduction measures than simply adding redundancy and overdesign to deal with component failures. The new techniques also provide a way to implement *safety-guided design* so that safety analysis guides the design generation rather than waiting until a design is complete to discover it is unsafe. Part III describes ways to use techniques based on STAMP to prevent accidents through system design, including design of the operating conditions and the safety management control structure.

STAMP can also be used to improve performance analysis. Performance monitoring of complex systems has created some dilemmas. Computers allow the collection of massive amounts of data, but analyzing that data to determine whether the system is moving toward the boundaries of safe behavior is difficult. The use of an accident model based on system theory and the basic concept of safety constraints may provide directions for identifying appropriate safety metrics and leading indicators; determining whether control over the safety constraints is adequate; evaluating the assumptions about the technical failures and potential design errors, organizational structure, and human behavior underlying the hazard analysis; detecting errors in the operational and environmental assumptions underlying the design and the organizational culture; and identifying any maladaptive changes over time that could increase risk of accidents to unacceptable levels.

Finally, STAMP points the way to very different approaches to risk assessment. Currently, risk assessment is firmly rooted in the probabilistic analysis of failure events. Attempts to extend current PRA techniques to software and other new technology, to management, and to cognitively complex human control activities have been disappointing. This way forward may lead to a dead end. Significant progress in risk assessment for complex systems will require innovative approaches starting from a completely different theoretical foundation.

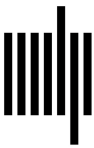
Engineering a Safer World

Systems Thinking Applied to Safety

By: Nancy G. Leveson

Citation:

Engineering a Safer World: Systems Thinking Applied to Safety
By: Nancy G. Leveson
DOI: 10.7551/mitpress/8179.001.0001
ISBN (electronic): 9780262298247
Publisher: The MIT Press
Published: 2016



The MIT Press

For information about special quantity discounts, please email special_sales@mitpress.mit.edu

This book was set in Syntax and Times Roman by Toppan Best-set Premedia Limited. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Leveson, Nancy.
Engineering a safer world : systems thinking applied to safety / Nancy G. Leveson.
p. cm.—(Engineering systems)
Includes bibliographical references and index.
ISBN 978-0-262-01662-9 (hardcover : alk. paper)
1. Industrial safety. 2. System safety. I. Title.
T55.L466 2012
620.8′6—dc23
2011014046

10 9 8 7 6 5 4 3 2 1