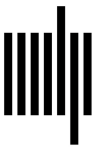
Engineering a Safer World

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

3 Systems Theory and Its Relationship to Safety

To achieve the goals set at the end of the last chapter, a new theoretical underpinning is needed for system safety. Systems theory provides that foundation. This chapter introduces some basic concepts in systems theory, how this theory is reflected in system engineering, and how all of this relates to system safety.

3.1 An Introduction to Systems Theory

Systems theory dates from the 1930s and 1940s and was a response to limitations of the classic analysis techniques in coping with the increasingly complex systems starting to be built at that time [36]. Norbert Wiener applied the approach to control and communications engineering [210], while Ludwig von Bertalanffy developed similar ideas for biology [21]. Bertalanffy suggested that the emerging ideas in various fields could be combined into a general theory of systems.

In the traditional scientific method, sometimes referred to as *divide and conquer*, systems are broken into distinct parts so that the parts can be examined separately: Physical aspects of systems are decomposed into separate physical components, while behavior is decomposed into discrete events over time.

Physical aspects → Separate physical components

Behavior → Discrete events over time

This decomposition (formally called *analytic reduction*) assumes that the separation is feasible: that is, each component or subsystem operates independently, and analysis results are not distorted when these components are considered separately. This assumption in turn implies that the components or events are not subject to feedback loops and other nonlinear interactions and that the behavior of the components is the same when examined singly as when they are playing their part in the whole. A third fundamental assumption is that the principles governing the assembling of the components into the whole are straightforward, that is, the interactions

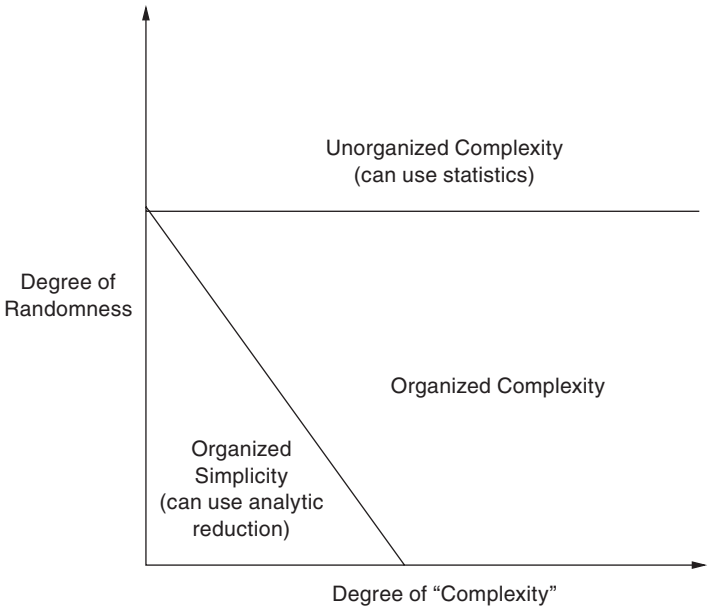


Figure 3.1
Three categories of systems (adapted from Gerald Weinberg, *An Introduction to General Systems Thinking* [John Wiley, 1975]).

among the subsystems are simple enough that they can be considered separate from the behavior of the subsystems themselves.

These are reasonable assumptions, it turns out, for many of the physical regularities of the universe. System theorists have described these systems as displaying *organized simplicity* (figure 3.1) [207]. Such systems can be separated into non-interacting subsystems for analysis purposes: the precise nature of the component interactions is known and interactions can be examined pairwise. Analytic reduction has been highly effective in physics and is embodied in structural mechanics.

Other types of systems display what systems theorists have labeled *unorganized complexity*—that is, they lack the underlying structure that allows reductionism to be effective. They can, however, often be treated as aggregates: They are complex, but regular and random enough in their behavior that they can be studied statistically. This study is simplified by treating them as a structureless mass with interchangeable parts and then describing them in terms of averages. The basis of this approach is the *law of large numbers*: The larger the population, the more likely that observed values are close to the predicted average values. In physics, this approach is embodied in statistical mechanics.

A third type of system exhibits what system theorists call *organized complexity*. These systems are too complex for complete analysis and too organized for statistics; the averages are deranged by the underlying structure [207]. Many of the complex engineered systems of the post–World War II era, as well as biological systems and social systems, fit into this category. Organized complexity also represents particularly well the problems that are faced by those attempting to build complex software, and it explains the difficulty computer scientists have had in attempting to apply analysis and statistics to software.

Systems theory was developed for this third type of system. The systems approach focuses on systems taken as a whole, not on the parts taken separately. It assumes that some properties of systems can be treated adequately only in their entirety, taking into account all facets relating the social to the technical aspects [161]. These system properties derive from the relationships between the parts of systems: how the parts interact and fit together [1]. Concentrating on the analysis and design of the whole as distinct from the components or parts provides a means for studying systems exhibiting organized complexity.

The foundation of systems theory rests on two pairs of ideas: (1) *emergence* and *hierarchy* and (2) *communication* and *control* [36].

3.2 Emergence and Hierarchy

A general model of complex systems can be expressed in terms of a *hierarchy* of levels of organization, each more complex than the one below, where a level is characterized by having *emergent* properties. Emergent properties do not exist at lower levels; they are meaningless in the language appropriate to those levels. The shape of an apple, although eventually explainable in terms of the cells of the apple, has no meaning at that lower level of description. The operation of the processes at the lower levels of the hierarchy result in a higher level of complexity—that of the whole apple itself—that has emergent properties, one of them being the apple's shape [36]. The concept of emergence is the idea that at a given level of complexity, some properties characteristic of that level (emergent at that level) are irreducible.

Hierarchy theory deals with the fundamental differences between one level of complexity and another. Its ultimate aim is to explain the relationships between different levels: what generates the levels, what separates them, and what links them. Emergent properties associated with a set of components at one level in a hierarchy are related to *constraints upon the degree of freedom* of those components. Describing the emergent properties resulting from the imposition of constraints requires a language at a higher level (a metalevel) different than that describing the components themselves. Thus, different languages of description are appropriate at different levels.

Reliability is a component property.[1] Conclusions can be reached about the reliability of a valve in isolation, where reliability is defined as the probability that the behavior of the valve will satisfy its specification over time and under given conditions.

Safety, on the other hand, is clearly an emergent property of systems: Safety can be determined only in the context of the whole. Determining whether a plant is acceptably safe is not possible, for example, by examining a single valve in the plant. In fact, statements about the "safety of the valve" without information about the context in which that valve is used are meaningless. Safety is determined by the relationship between the valve and the other plant components. As another example, pilot procedures to execute a landing might be safe in one aircraft or in one set of circumstances but unsafe in another.

Although they are often confused, reliability and safety are different properties. The pilots may reliably execute the landing procedures on a plane or at an airport in which those procedures are unsafe. A gun when discharged out on a desert with no other humans or animals for hundreds of miles may be both safe and reliable. When discharged in a crowded mall, the reliability will not have changed, but the safety most assuredly has.

Because safety is an emergent property, it is not possible to take a single system component, like a software module or a single human action, in isolation and assess its safety. A component that is perfectly safe in one system or in one environment may not be when used in another.

The new model of accidents introduced in part II of this book incorporates the basic systems theory idea of hierarchical levels, where constraints or lack of constraints at the higher levels control or allow lower-level behavior. Safety is treated as an emergent property at each of these levels. Safety depends on the enforcement of constraints on the behavior of the components in the system, including constraints on their potential interactions. Safety in the batch chemical reactor in the previous chapter, for example, depends on the enforcement of a constraint on the relationship between the state of the catalyst valve and the water valve.

3.3 Communication and Control

The second major pair of ideas in systems theory is *communication* and *control.* An example of regulatory or *control* action is the imposition of *constraints* upon the

1. This statement is somewhat of an oversimplification, because the reliability of a system component can, under some conditions (e.g., magnetic interference or excessive heat) be impacted by its environment. The basic reliability of the component, however, can be defined and measured in isolation, whereas the safety of an individual component is undefined except in a specific environment.

activity at one level of a hierarchy, which define the "laws of behavior" at that level. Those laws of behavior yield activity meaningful at a higher level. Hierarchies are characterized by control processes operating at the interfaces between levels [36].

The link between control mechanisms studied in natural systems and those engineered in man-made systems was provided by a part of systems theory known as cybernetics. Checkland writes:

> Control is always associated with the imposition of constraints, and an account of a control process necessarily requires our taking into account at least two hierarchical levels. At a given level, it is often possible to describe the level by writing dynamical equations, on the assumption that one particle is representative of the collection and that the forces at other levels do not interfere. But any description of a control process entails an upper level imposing constraints upon the lower. The upper level is a source of an alternative (simpler) description of the lower level in terms of specific functions that are emergent as a result of the imposition of constraints [36, p. 87].

Note Checkland's statement about control always being associated with the imposition of constraints. Imposing *safety constraints* plays a fundamental role in the approach to safety presented in this book. The limited focus on avoiding failures, which is common in safety engineering today, is replaced by the larger concept of imposing constraints on system behavior to avoid unsafe events or conditions, that is, hazards.

Control in open systems (those that have inputs and outputs from their environment) implies the need for *communication.* Bertalanffy distinguished between *closed systems*, in which unchanging components settle into a state of equilibrium, and *open systems*, which can be thrown out of equilibrium by exchanges with their environment.

In control theory, open systems are viewed as interrelated components that are kept in a state of dynamic equilibrium by feedback loops of information and control. The plant's overall performance has to be controlled in order to produce the desired product while satisfying cost, safety, and general quality constraints.

In order to control a process, four conditions are required [10]:

- *Goal Condition:* The controller must have a goal or goals (for example, to maintain the setpoint).
- *Action Condition:* The controller must be able to affect the state of the system. In engineering, control actions are implemented by *actuators.*
- *Model Condition:* The controller must be (or contain) a model of the system (see section 4.3).
- *Observability Condition:* The controller must be able to ascertain the state of the system. In engineering terminology, observation of the state of the system is provided by *sensors.*

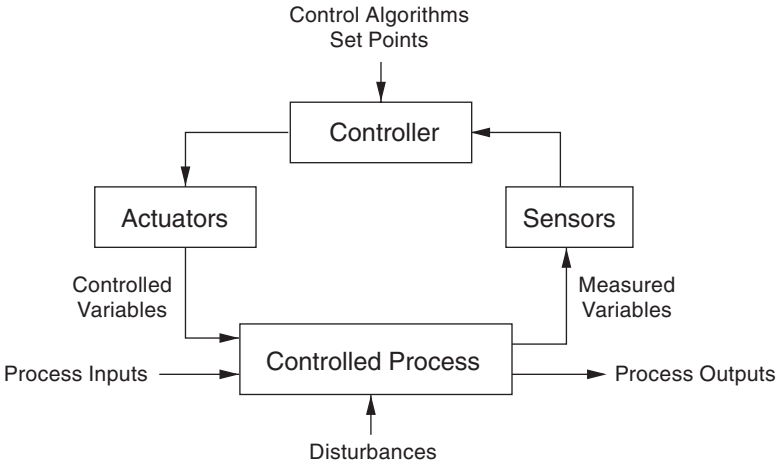


Figure 3.2
A standard control loop.

Figure 3.2 shows a typical control loop. The plant controller obtains information about (observes) the process state from measured variables (*feedback*) and uses this information to initiate action by manipulating *controlled variables* to keep the process operating within predefined limits or *set points* (the goal) despite disturbances to the process. In general, the maintenance of any open-system hierarchy (either biological or man-made) will require a set of processes in which there is communication of information for regulation or control [36].

Control actions will generally lag in their effects on the process because of delays in signal propagation around the control loop: an actuator may not respond immediately to an external command signal (called *dead time*); the process may have delays in responding to manipulated variables (*time constants*); and the sensors may obtain values only at certain sampling intervals (*feedback delays*). Time lags restrict the speed and extent with which the effects of disturbances, both within the process itself and externally derived, can be reduced. They also impose extra requirements on the controller, for example, the need to infer delays that are not directly observable.

The model condition plays an important role in accidents and safety. In order to create effective control actions, the controller must know the current state of the controlled process and be able to estimate the effect of various control actions on that state. As discussed further in section 4.3, many accidents have been caused by the controller incorrectly assuming the controlled system was in a particular state and imposing a control action (or not providing one) that led to a loss: the Mars Polar Lander descent engine controller, for example, assumed that the spacecraft

was on the surface of the planet and shut down the descent engines. The captain of the *Herald of Free Enterprise* thought the car deck doors were shut and left the mooring.

3.4 Using Systems Theory to Understand Accidents

Safety approaches based on systems theory consider accidents as arising from the interactions among system components and usually do not specify single causal variables or factors [112]. Whereas industrial (occupational) safety models and event chain models focus on unsafe acts or conditions, classic system safety models instead look at what went wrong with the system's operation or organization to allow the accident to take place.

This systems approach treats safety as an emergent property that arises when the system components interact within an environment. Emergent properties like safety are controlled or enforced by a set of constraints (control laws) related to the behavior of the system components. For example, the spacecraft descent engines must remain on until the spacecraft reaches the surface of the planet and the car deck doors on the ferry must be closed before leaving port. Accidents result from interactions among components that violate these constraints—in other words, from a lack of appropriate constraints on the interactions. Component interaction accidents, as well as component failure accidents, can be explained using these concepts.

Safety then can be viewed as a control problem. Accidents occur when component failures, external disturbances, and/or dysfunctional interactions among system components are not adequately controlled. In the space shuttle *Challenger* loss, the O-rings did not adequately control propellant gas release by sealing a tiny gap in the field joint. In the Mars Polar Lander loss, the software did not adequately control the descent speed of the spacecraft—it misinterpreted noise from a Hall effect sensor (feedback of a measured variable) as an indication the spacecraft had reached the surface of the planet. Accidents such as these, involving engineering design errors, may in turn stem from inadequate control over the development process. A Milstar satellite was lost when a typo in the software load tape was not detected during the development and testing. Control is also imposed by the management functions in an organization—the *Challenger* and *Columbia* losses, for example, involved inadequate controls in the launch-decision process.

While events reflect the *effects* of dysfunctional interactions and inadequate enforcement of safety constraints, the inadequate control itself is only indirectly reflected by the events—the events are the *result* of the inadequate control. The control structure itself must be examined to determine why it was inadequate to maintain the constraints on safe behavior and why the events occurred.

As an example, the unsafe behavior (hazard) in the *Challenger* loss was the release of hot propellant gases from the field joint. The miscreant O-ring was used to control the hazard—that is, its role was to seal a tiny gap in the field joint created by pressure at ignition. The loss occurred because the system design, including the O-ring, did not effectively impose the required constraint on the propellant gas release. Starting from here, there are then several questions that need to be answered to understand why the accident occurred and to obtain the information necessary to prevent future accidents. Why was this particular design unsuccessful in imposing the constraint, why was it chosen (what was the decision process), why was the flaw not found during development, and was there a different design that might have been more successful? These questions and others consider the original *design process*.

Understanding the accident also requires examining the contribution of the *operations process*. Why were management decisions made to launch despite warnings that it might not be safe to do so? One constraint that was violated during operations was the requirement to correctly handle feedback about any potential violation of the safety design constraints, in this case, feedback during operations that the control by the O-rings of the release of hot propellant gases from the field joints was not being adequately enforced by the design. There were several instances of feedback that was not adequately handled, such as data about O-ring blowby and erosion during previous shuttle launches and feedback by engineers who were concerned about the behavior of the O-rings in cold weather. Although the lack of redundancy provided by the second O-ring was known long before the loss of *Challenger*, that information was never incorporated into the NASA Marshall Space Flight Center database and was unknown by those making the launch decision. In addition, there was missing feedback about changes in the design and testing procedures during operations, such as the use of a new type of putty and the introduction of new O-ring leak checks without adequate verification that they satisfied system safety constraints on the field joints. As a final example, the control processes that ensured unresolved safety concerns were fully considered before each flight, that is, the flight readiness reviews and other feedback channels to project management making flight decisions, were flawed.

Systems theory provides a much better foundation for safety engineering than the classic analytic reduction approach underlying event-based models of accidents. It provides a way forward to much more powerful and effective safety and risk analysis and management procedures that handle the inadequacies and needed extensions to current practice described in chapter 2.

Combining a systems-theoretic approach to safety with system engineering processes will allow designing safety into the system as it is being developed or reengineered. System engineering provides an appropriate vehicle for this process

because it rests on the same systems theory foundation and involves engineering the system as a whole.

3.5 Systems Engineering and Safety

The emerging theory of systems, along with many of the historical forces noted in chapter 1, gave rise after World War II to a new emphasis in engineering, eventually called systems engineering. During and after the war, technology expanded rapidly and engineers were faced with designing and building more complex systems than had been attempted previously. Much of the impetus for the creation of this new discipline came from military programs in the 1950s and 1960s, particularly intercontinental ballistic missile (ICBM) systems. *Apollo* was the first nonmilitary government program in which systems engineering was recognized from the beginning as an essential function [24].

System Safety, as defined in MIL-STD-882, is a subdiscipline of system engineering. It was created at the same time and for the same reasons. The defense community tried using the standard safety engineering techniques on their complex new systems, but the limitations became clear when interface and component interaction problems went unnoticed until it was too late, resulting in many losses and near misses. When these early aerospace accidents were investigated, the causes of a large percentage of them were traced to deficiencies in design, operations, and management. Clearly, big changes were needed. System engineering along with its subdiscipline, System Safety, were developed to tackle these problems.

Systems theory provides the theoretical foundation for systems engineering, which views each system as an integrated whole even though it is composed of diverse, specialized components. The objective is to integrate the subsystems into the most effective system possible to achieve the overall objectives, given a prioritized set of design criteria. Optimizing the system design often requires making tradeoffs between these design criteria (goals).

The development of systems engineering as a discipline enabled the solution of enormously more complex and difficult technological problems than previously [137]. Many of the elements of systems engineering can be viewed merely as good engineering: It represents more a shift in emphasis than a change in content. In addition, while much of engineering is based on technology and science, systems engineering is equally concerned with overall management of the engineering process.

A systems engineering approach to safety starts with the basic assumption that some properties of systems, in this case safety, can only be treated adequately in the context of the social and technical system as a whole. A basic assumption of systems engineering is that optimization of individual components or subsystems will not in

general lead to a system optimum; in fact, improvement of a particular subsystem may actually worsen the overall system performance because of complex, nonlinear interactions among the components. When each aircraft tries to optimize its path from its departure point to its destination, for example, the overall air transportation system throughput may not be optimized when they all arrive at a popular hub at the same time. One goal of the air traffic control system is to optimize the overall air transportation system throughput while, at the same time, trying to allow as much flexibility for the individual aircraft and airlines to achieve their goals. In the end, if system engineering is successful, everyone gains. Similarly, each pharmaceutical company acting to optimize its profits, which is a legitimate and reasonable company goal, will not necessarily optimize the larger societal *system* goal of producing safe and effective pharmaceutical and biological products to enhance public health. These system engineering principles are applicable even to systems beyond those traditionally thought of as in the engineering realm. The financial system and its meltdown starting in 2007 is an example of a social system that could benefit from system engineering concepts.

Another assumption of system engineering is that individual component behavior (including events or actions) cannot be understood without considering the components' role and interaction within the system as a whole. This basis for systems engineering has been stated as the principle that a system is more than the sum of its parts. Attempts to improve long-term safety in complex systems by analyzing and changing individual components have often proven to be unsuccessful over the long term. For example, Rasmussen notes that over many years of working in the field of nuclear power plant safety, he found that attempts to improve safety from models of local features were compensated for by people adapting to the change in an unpredicted way [167].

Approaches used to enhance safety in complex systems must take these basic systems engineering principles into account. Otherwise, our safety engineering approaches will be limited in the types of accidents and systems they can handle. At the same time, approaches that include them, such as those described in this book, have the potential to greatly improve our ability to engineer safer and more complex systems.

3.6 Building Safety into the System Design

System Safety, as practiced by the U.S. defense and aerospace communities as well as the new approach outlined in this book, fit naturally within the general systems engineering process and the problem-solving approach that a system view provides. This problem-solving process entails several steps. First, a need or problem is specified in terms of objectives that the system must satisfy along with criteria that can

be used to rank alternative designs. For a system that has potential hazards, the objectives will include safety objectives and criteria along with high-level requirements and safety design constraints. The hazards for an automated train system, for example, might include the train doors closing while a passenger is in the doorway. The safety-related design constraint might be that obstructions in the path of a closing door must be detected and the door closing motion reversed.

After the high-level requirements and constraints on the system design are identified, a process of system synthesis takes place that results in a set of alternative designs. Each of these alternatives is analyzed and evaluated in terms of the stated objectives and design criteria, and one alternative is selected to be implemented. In practice, the process is highly iterative: The results from later stages are fed back to early stages to modify objectives, criteria, design alternatives, and so on. Of course, the process described here is highly simplified and idealized.

The following are some examples of basic systems engineering activities and the role of safety within them:

- *Needs analysis:* The starting point of any system design project is a perceived need. This need must first be established with enough confidence to justify the commitment of resources to satisfy it and understood well enough to allow appropriate solutions to be generated. Criteria must be established to provide a means to evaluate both the evolving and final system. If there are hazards associated with the operation of the system, safety should be included in the needs analysis.
- *Feasibility studies:* The goal of this step in the design process is to generate a set of realistic designs. This goal is accomplished by identifying the principal constraints and design criteria—including safety constraints and safety design criteria—for the specific problem being addressed and then generating plausible solutions to the problem that satisfy the requirements and constraints and are physically and economically feasible.
- *Trade studies:* In trade studies, the alternative feasible designs are evaluated with respect to the identified design criteria. A hazard might be controlled by any one of several safeguards: A trade study would determine the relative desirability of each safeguard with respect to effectiveness, cost, weight, size, safety, and any other relevant criteria. For example, substitution of one material for another may reduce the risk of fire or explosion, but may also reduce reliability or efficiency. Each alternative design may have its own set of safety constraints (derived from the system hazards) as well as other performance goals and constraints that need to be assessed. Although decisions ideally should be based upon mathematical analysis, quantification of many of the key factors is often difficult, if not impossible, and subjective judgment often has to be used.

- *System architecture development and analysis:* In this step, the system engineers break down the system into a set of subsystems, together with the functions and constraints, including safety constraints, imposed upon the individual subsystem designs, the major system interfaces, and the subsystem interface topology. These aspects are analyzed with respect to desired system performance characteristics and constraints (again including safety constraints) and the process is iterated until an acceptable system design results. The preliminary design at the end of this process must be described in sufficient detail that subsystem implementation can proceed independently.
- *Interface analysis:* The interfaces define the functional boundaries of the system components. From a management standpoint, interfaces must (1) optimize visibility and control and (2) isolate components that can be implemented independently and for which authority and responsibility can be delegated [158]. From an engineering standpoint, interfaces must be designed to separate independent functions and to facilitate the integration, testing, and operation of the overall system. One important factor in designing the interfaces is safety, and safety analysis should be a part of the system interface analysis. Because interfaces tend to be particularly susceptible to design error and are implicated in the majority of accidents, a paramount goal of interface design is simplicity. Simplicity aids in ensuring that the interface can be adequately designed, analyzed, and tested prior to integration and that interface responsibilities can be clearly understood.

Any specific realization of this general systems engineering process depends on the engineering models used for the system components and the desired system qualities. For safety, the models commonly used to understand why and how accidents occur have been based on events, particularly failure events, and the use of reliability engineering techniques to prevent them. Part II of this book further details the alternative systems approach to safety introduced in this chapter, while part III provides techniques to perform many of these safety and system engineering activities.

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