

Engineering a Safer and More Secure World

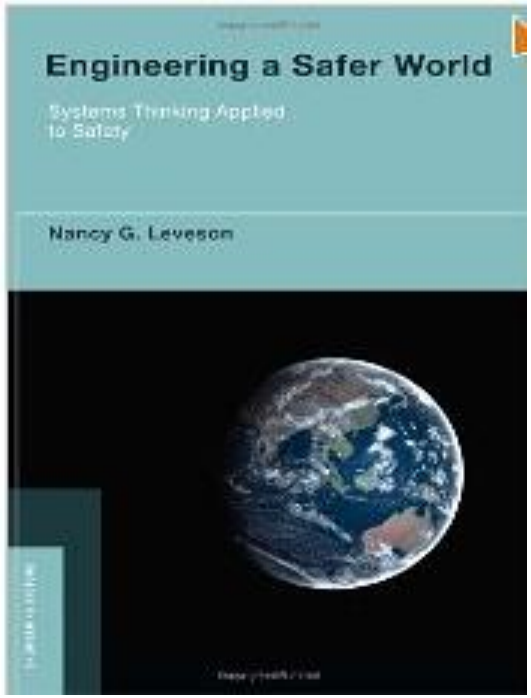
Nancy Leveson

MIT



Nancy Leveson, *Engineering a Safer World:*

Systems Thinking Applied to Safety



MIT Press, January 2012

Syllabus

DAY 1

- Why do accidents occur?
- Components of safety engineering
- Traditional safety engineering
  - Accident models (assumptions about why accidents occur)
  - Traditional Analysis techniques
- Why do we need something new?
- A systems-theoretic approach to safety engineering
- Introduction to CAST (accident causal analysis)

Syllabus

DAY 2

- Introduction to STPA (System Theoretic Process Analysis)
- Evaluations (Does this new approach work?)
- The way forward

General Definition of "Safety"

- Accident = Mishap = Loss: Any undesired and unplanned event that results in a loss
  - e.g., loss of human life or injury, property damage, environmental pollution, mission loss, negative business impact (damage to reputation, etc.), product launch delay, legal entanglements, etc. [MIL-STD-882]
  - Includes inadvertent and intentional losses (security)
- System goals vs. constraints (limits on how can achieve the goals)
- Safety: Absence of losses

What do you think is the cause of most accidents?

Why do Accidents Occur?

Watch the following video and try to answer the questions:

1. What did the official accident investigation conclude was the cause?

2. What other causal factors did you notice?

3. What was the “root cause” of the accident?

Uberlingen Mid-Air Collision

What were some of the causal factors you noticed in the Uberlingen accident?

Uberlingen Accident Factors

Official Report: Two major causes

1. Peter was too late in noticing the potential for the collision

2. Russian crew was wrong to obey ATC and not TCAS

(Human voice sounded more urgent)

Other Factors:

1. Only one controller on duty

a. Had to control two screens and switch between them

b. It was midnight, traffic was light, it had become standard for one controller to sleep in the duty room while the other handled traffic (management knew this was occurring)

c. Two other flights required Peter's attention at the time

d. Not able to hand off to another ATC as was the usual procedure because phones were not working

Uberlingen Accident Factors (2)

2. Planned maintenance (a software upgrade)
    a. Management did not plan for handling potential hazards created by maintenance activity
    b. Management did not inform controllers about the hazards and the potential for their tools (including collision alerts) not to work or to work more slowly during the upgrade

3. A controller at another control facility noticed the potential collision
    a. Could not phone Zurich because phones dead
    b. Not allowed to contact the planes directly.

Uberlingen Accident Factors (3)

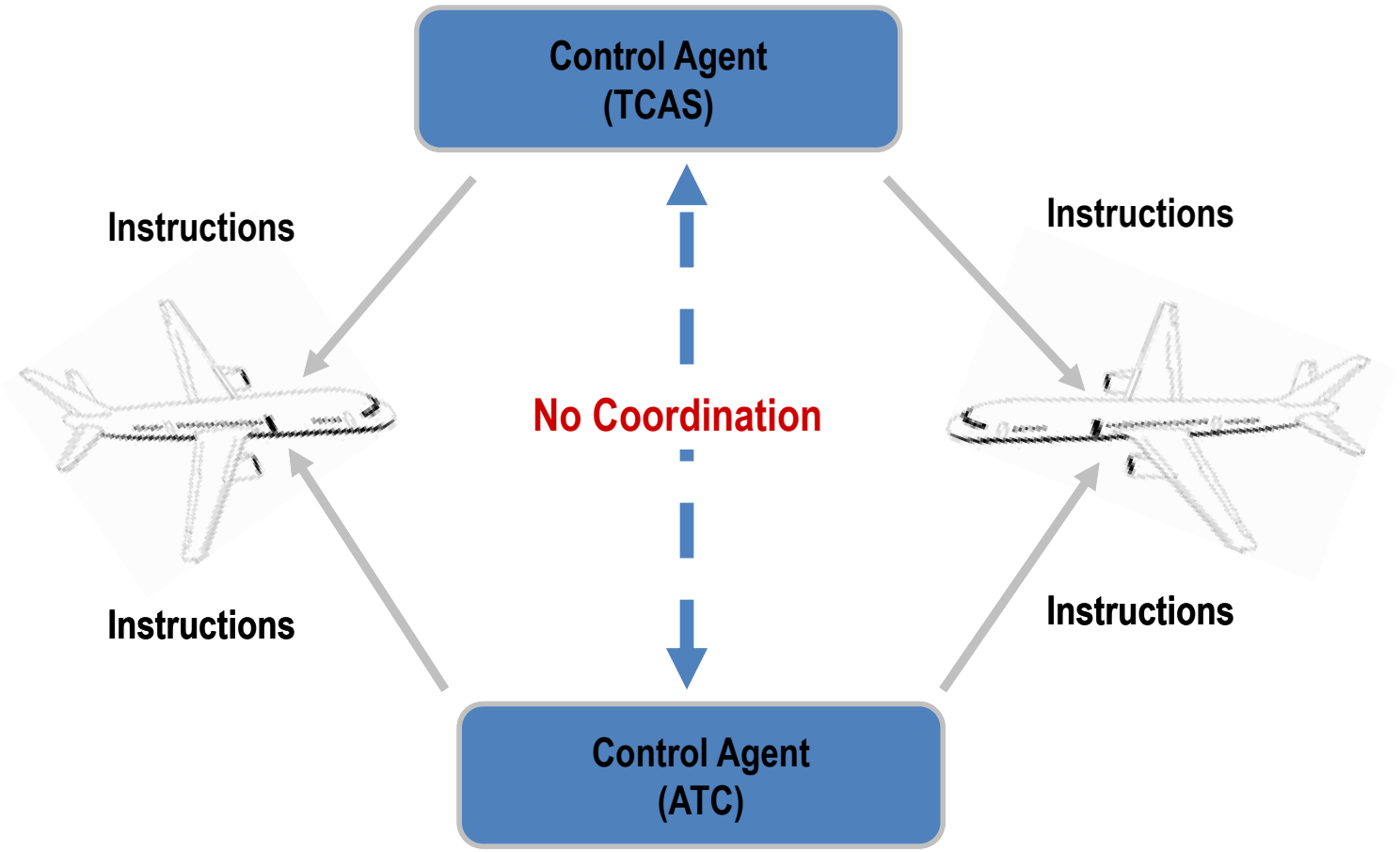
4. DHL pilot goes to washroom. TCAS alert has not sounded yet.
    a. By the rules, pilot flying must obey TCAS first
    b. Pilot not flying must call and report TCAS alert to controller (no direct downlink of alert technically possible at time TCAS created).
    c. Radio frequency is blocked

5. No TCAS reversal because did not meet conditions

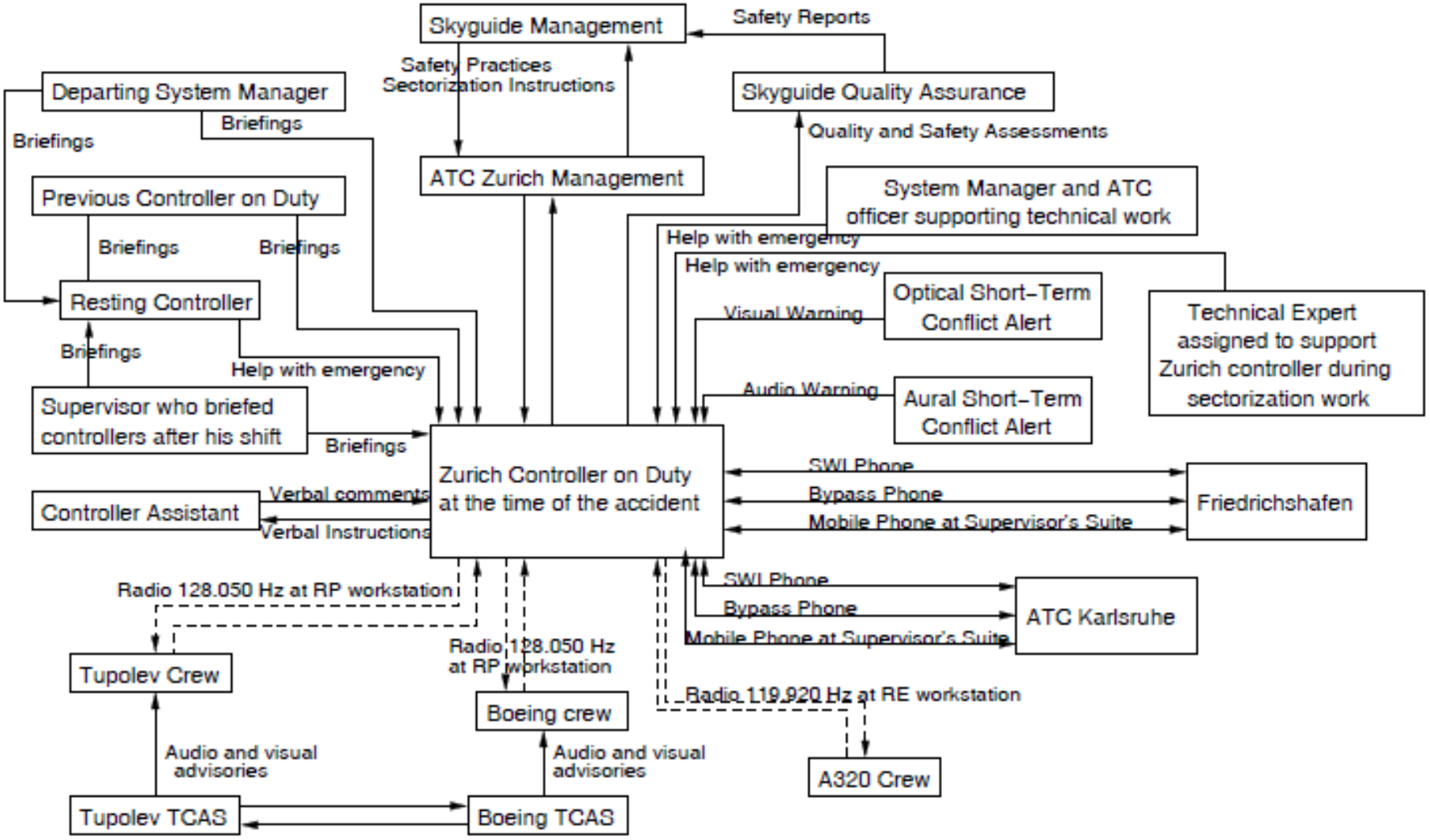
6. DHL crew followed TCAS, Russian crew followed ATC
    a. Rule in West is to follow TCAS if conflict
    b. No hard and fast rules in Russia if there is a conflict. Russian pilots used to following ATC (trusted more).
        i. TCAS pilot's guide ambiguous
        ii. TU-154 Ops Manual contained conflicting information
    c. More urgent human voice vs. machine-generated voice

Uncoordinated “Control Agents”

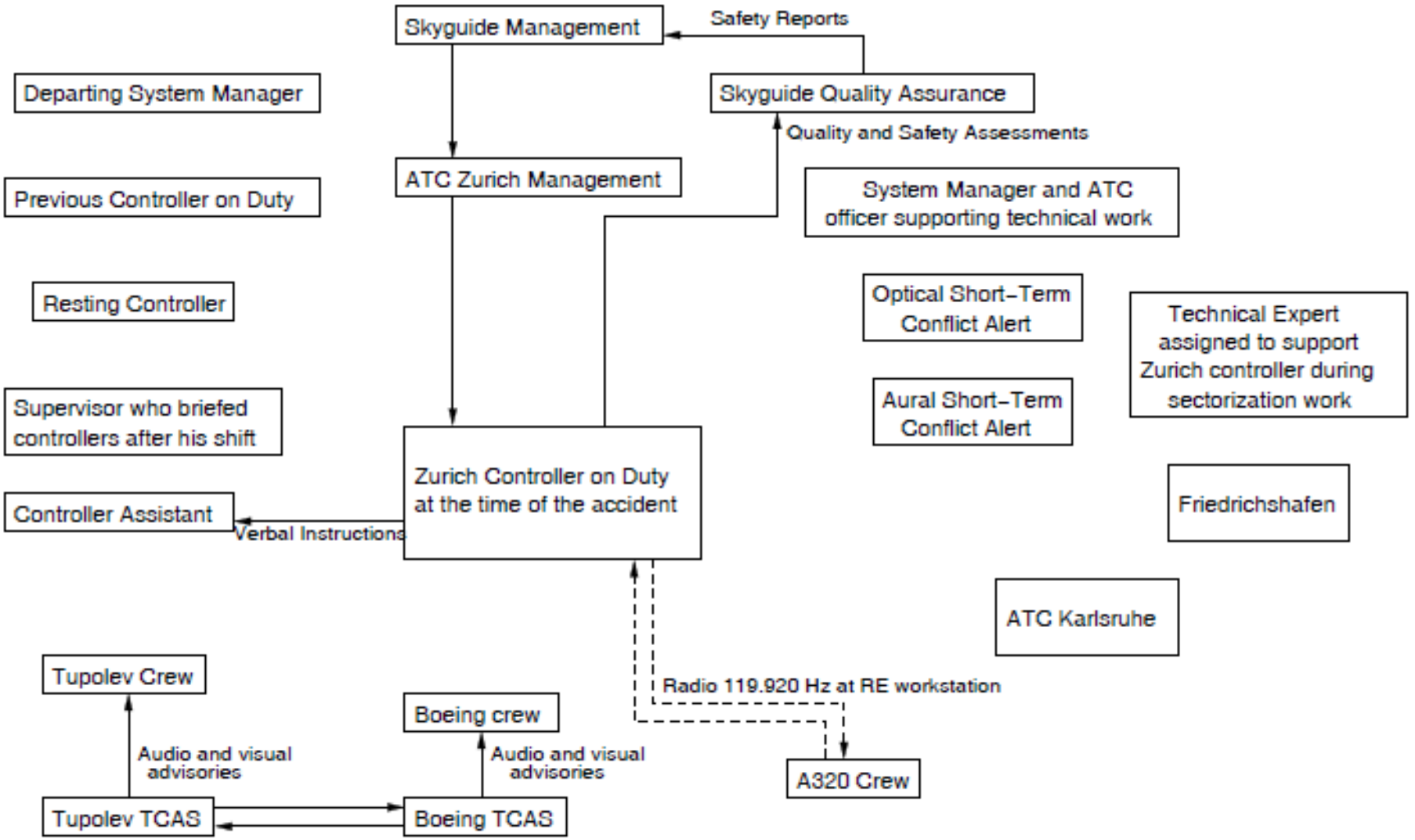
“UNSAFE STATE”
BOTH TCAS and ATC provide uncoordinated & independent instructions



Communication Links Theoretically in Place in Uberlingen Accident



Communication Links Actually in Place



Some additional factors not in the video

- A year prior there was a near miss due to conflicting TCAS and ATC commands
  - Two Japanese airliners
  - One pilot made evasive maneuvers based on visual judgement.
    - Aircraft came within 300 ft
    - Evasive maneuvers caused ~100 injuries
  - Japan called for changes, but ICAO did not take action until after Uberlingen
- Four other near misses in Europe before Uberlingen collision (involving one flight crew obeying TCAS and one following the air traffic controller)

Uberlingen continued

- TCAS Pilot's Guide was ambiguous about TCAS / ATC precedence
- Tu-154 Flight Operations Manual had contradictory sections
  - Chapter 8.18.3.2 forbids maneuvers contrary to TCAS
  - Chapter 8.18.3.4 says "most important tool" is executing ATC instructions. TCAS described as an additional instrument.

Components of Safety Engineering

- Investigating accidents (learning from events)
- Preventing Accidents
  - Hazard Analysis
  - Design for Safety
- Operations
- Management
- Risk Assessment?

Traditional Safety Engineering

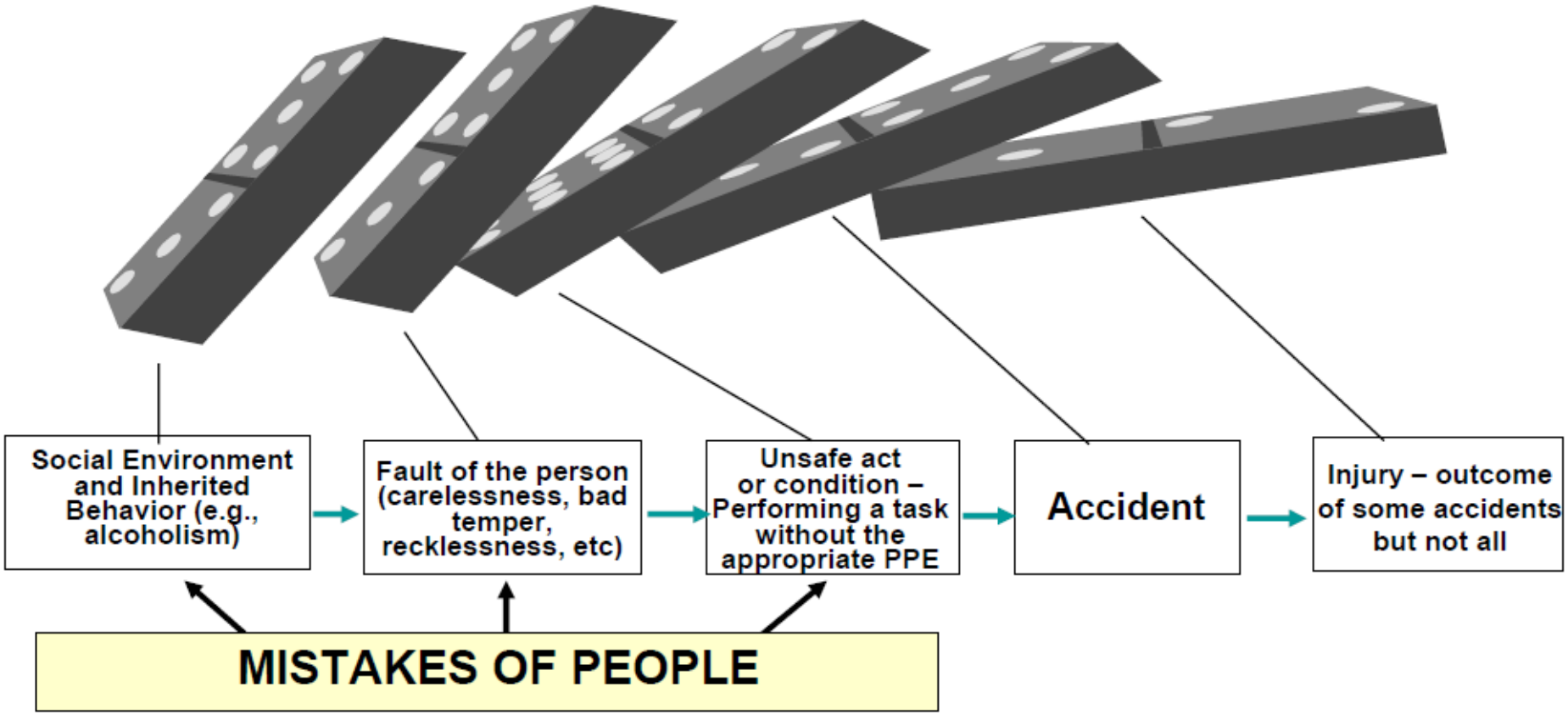
Accident Causality Models

- Underlie all our efforts to engineer for safety
- Explain why accidents occur
- Determine the way we prevent and investigate accidents
- May not be aware you are using one, but you are
- Imposes patterns on accidents

"All models are wrong, some models are useful"

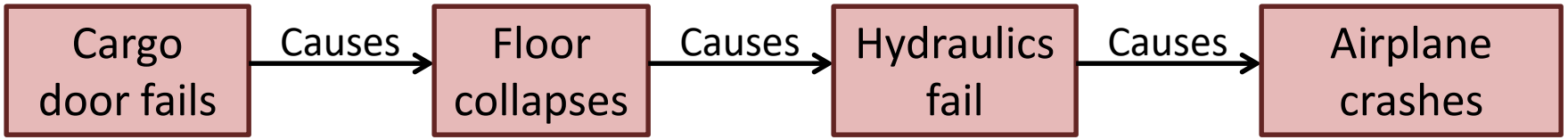
George Box

Heinrich's Domino Model of Accident Causation (1932)



Domino “Chain of events” Model

DC-10:

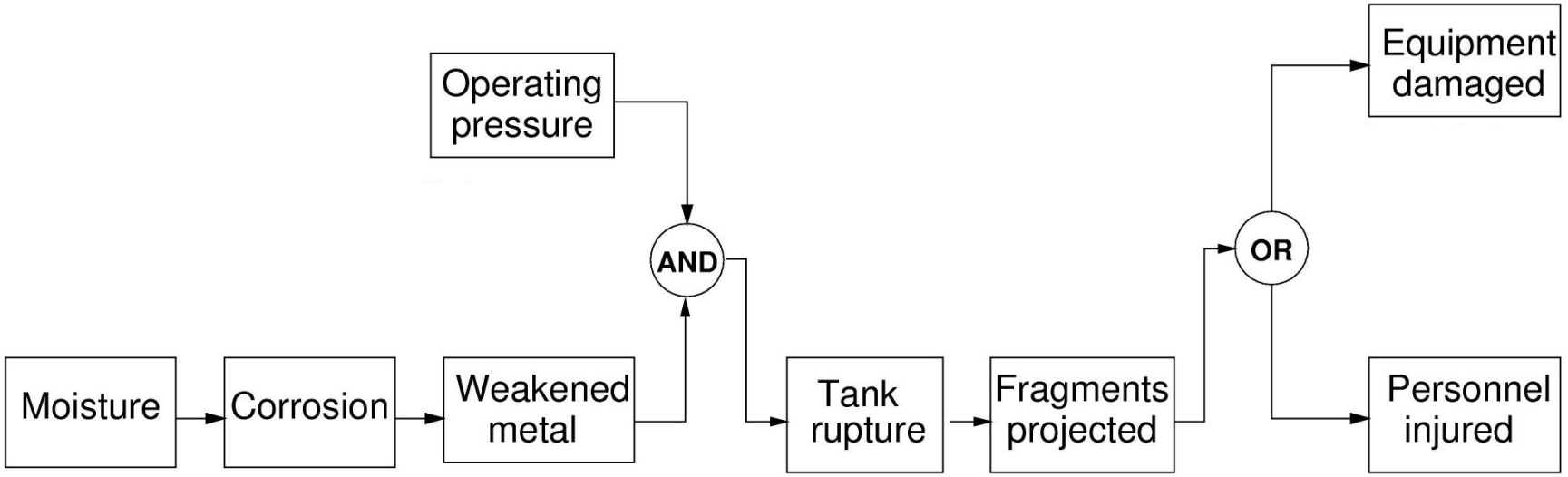


Event-based

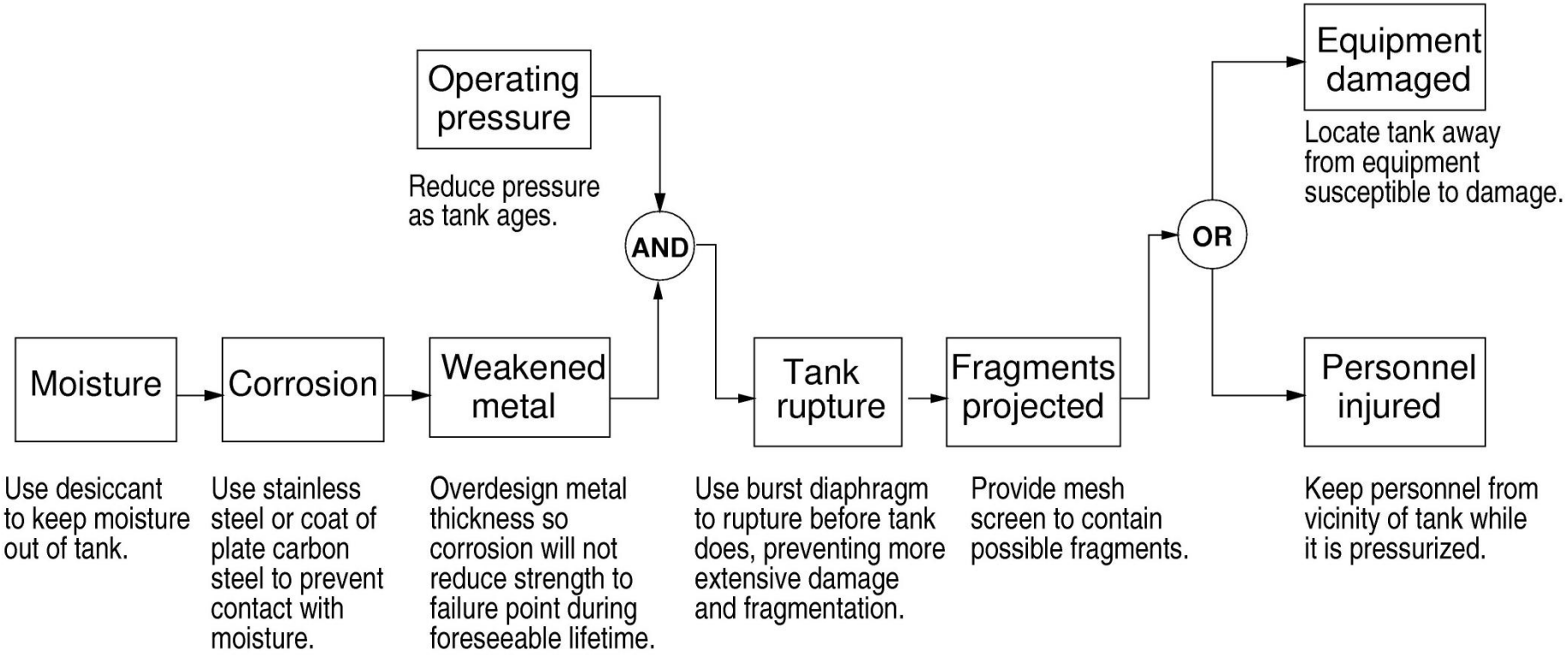
The Domino Model in action

John Thomas

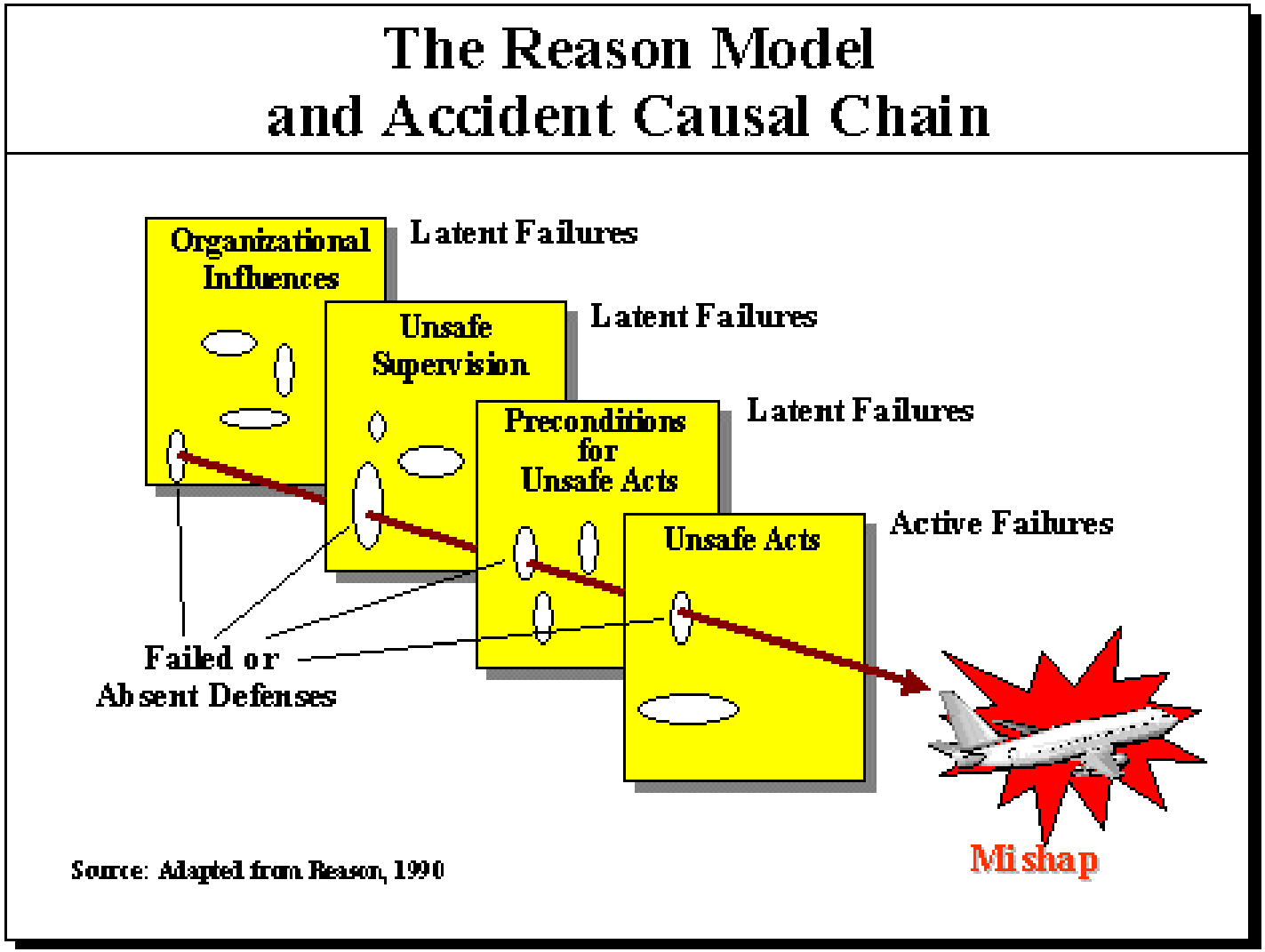
Chain-of-events example

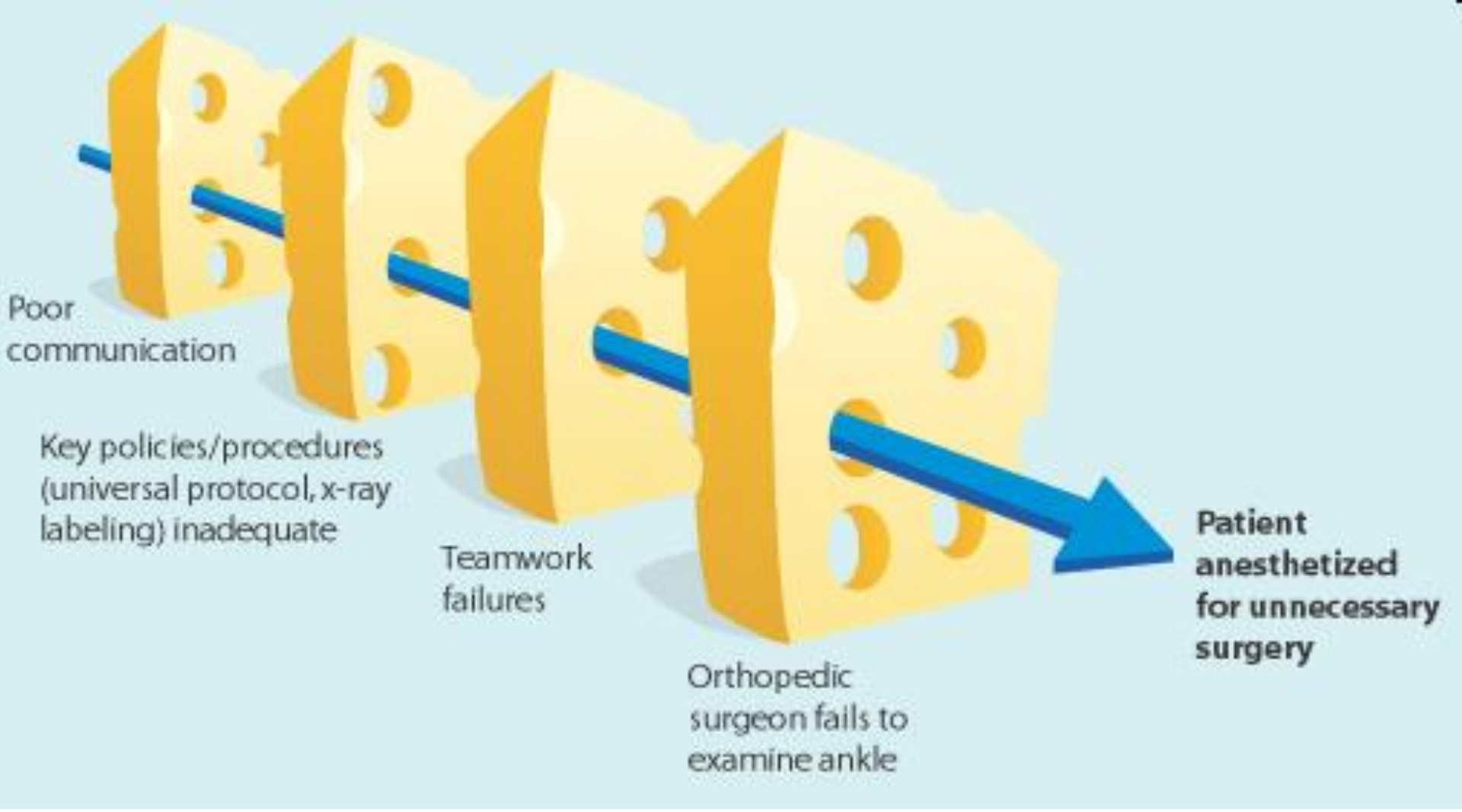


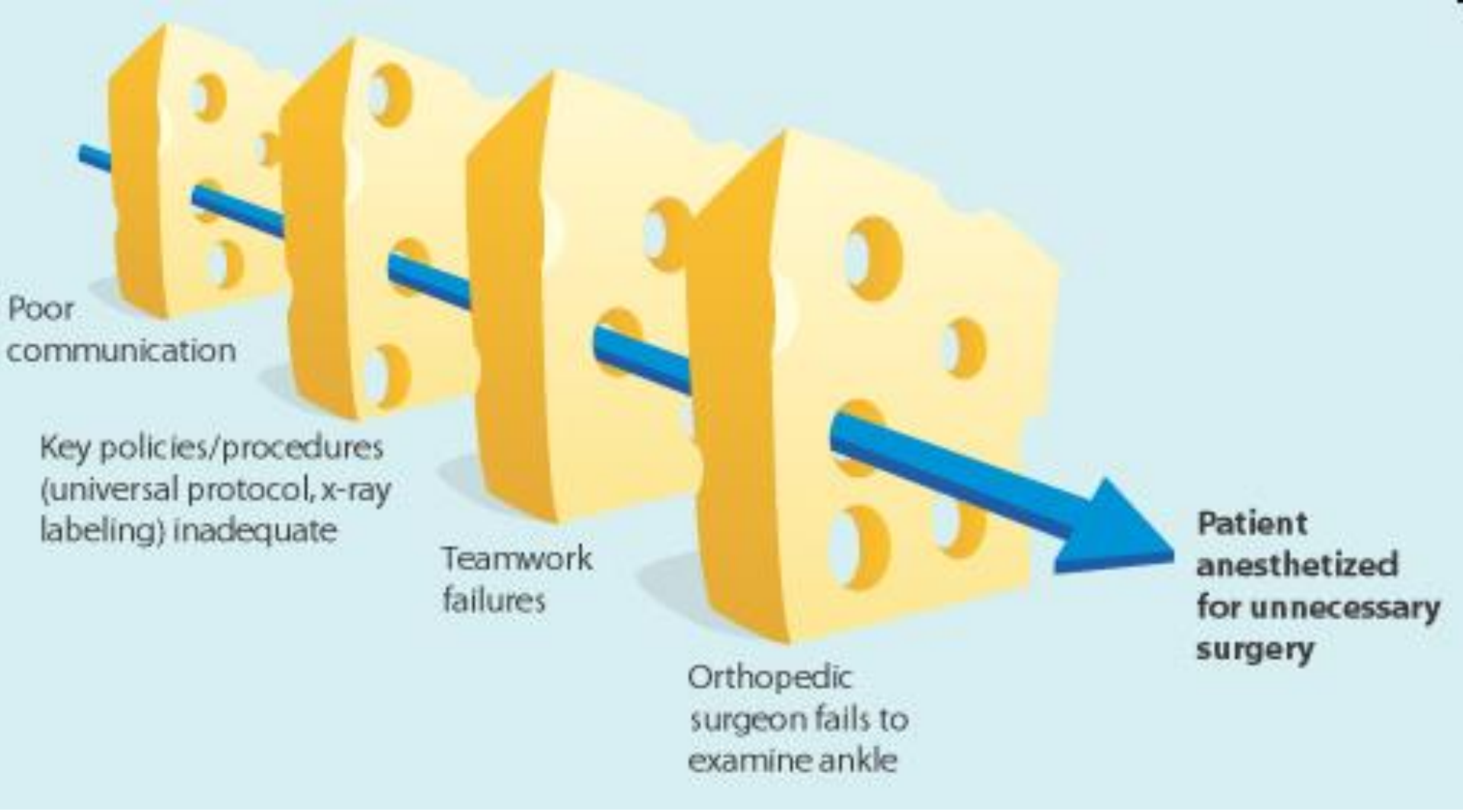
Chain-of-events example



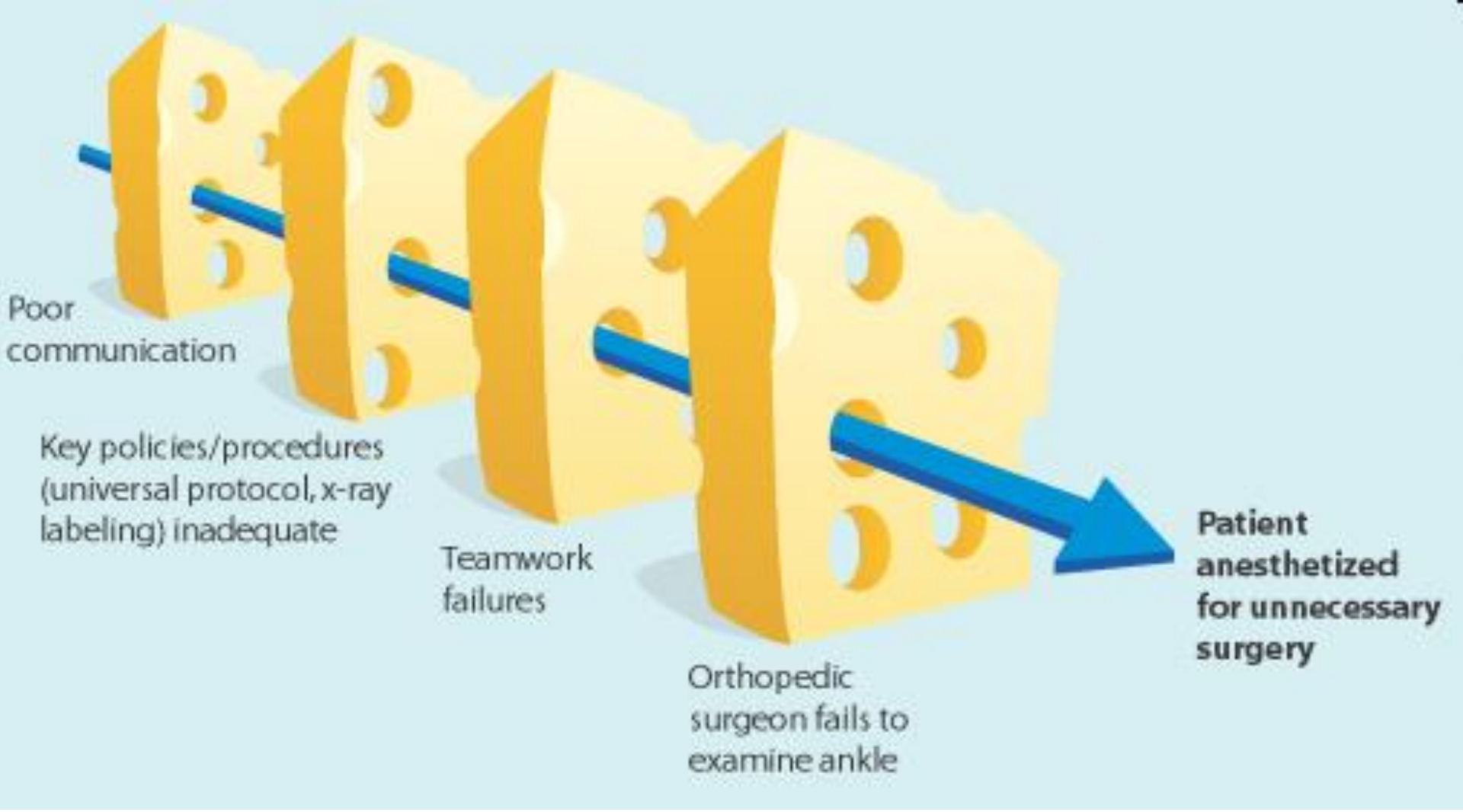
Reason Swiss Cheese = Domino Model



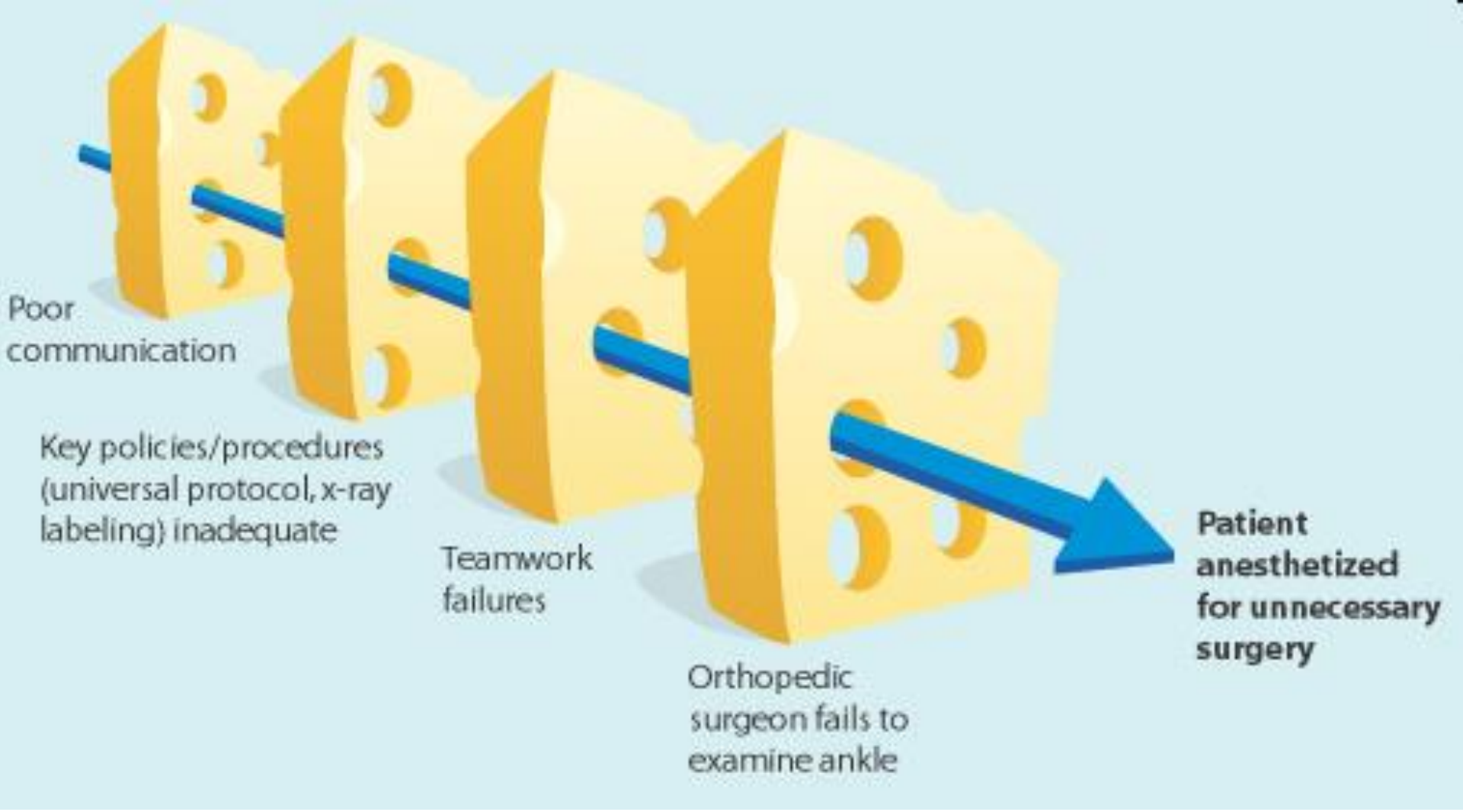

Poor communication
Key policies/procedures (universal protocol, x-ray labeling) inadequate
Teamwork failures
Orthopedic surgeon fails to examine ankle
Patient anesthetized for unnecessary surgery



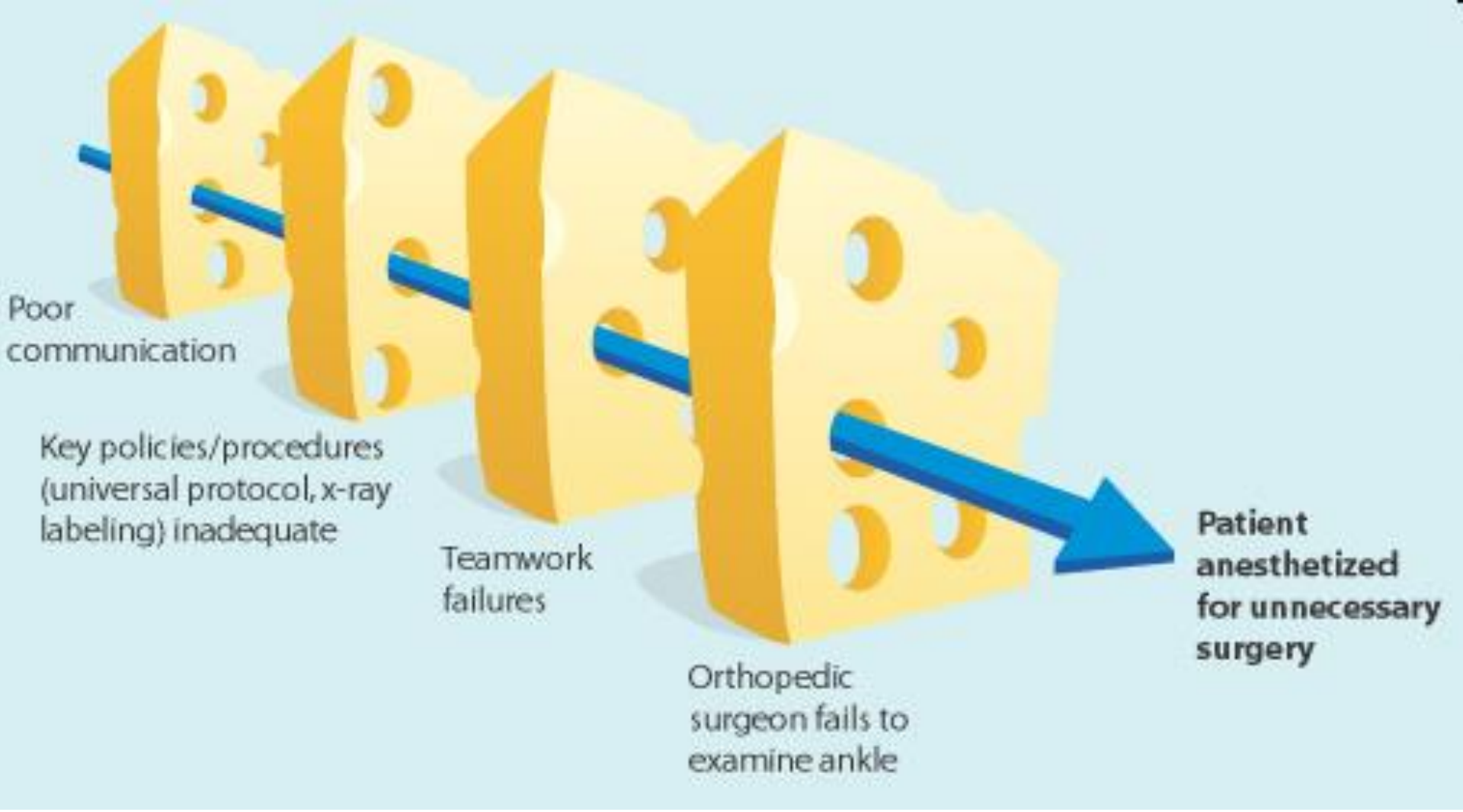
Ignores common cause failures of defenses (systemic accident factors)



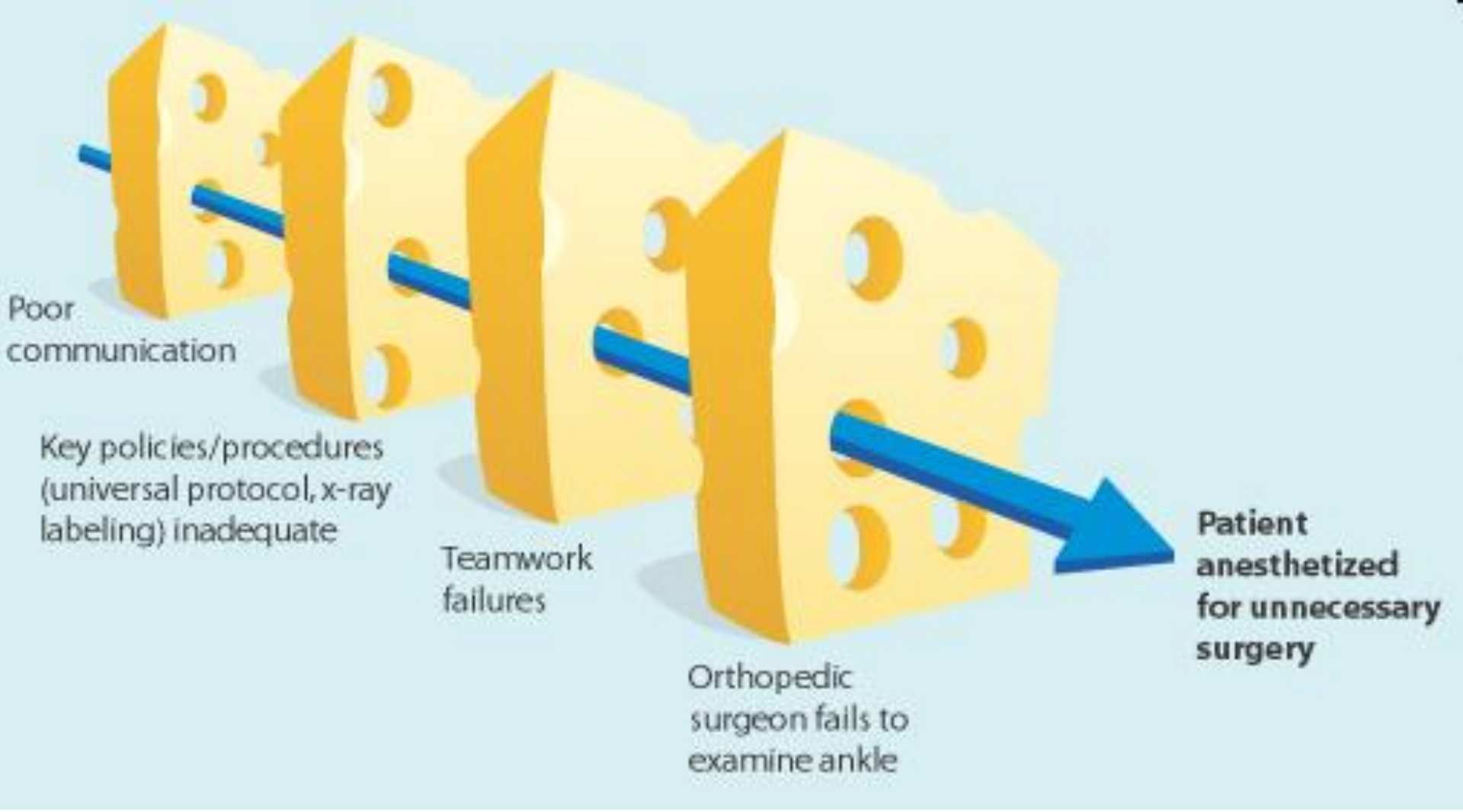
Assumes accidents are random events coming together accidentally



Assumes some (linear) causality or precedence in the cheese slices (and holes)



Does not include migration to states of higher risk

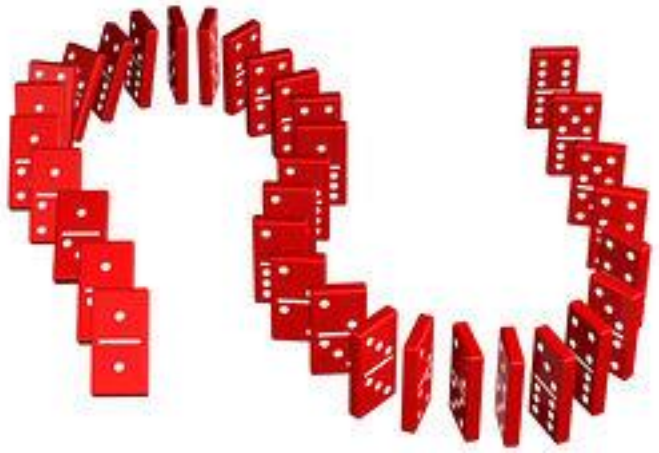


Just a chain of events, no explanation of "why" events occurred

How do you find the chain of events before an accident occurs?

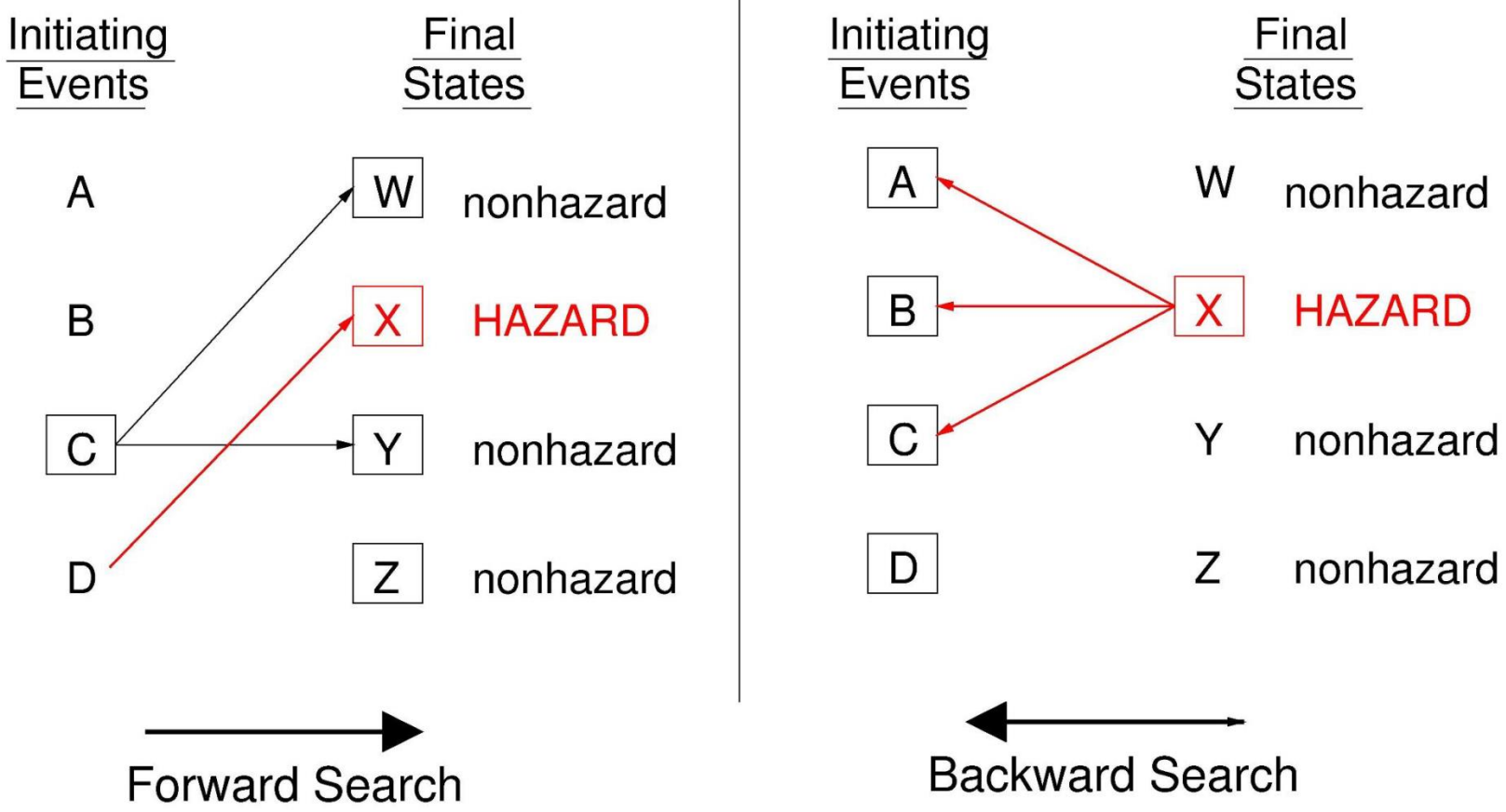
Hazard Analysis

Traditional Safety Methods Based on Chain-of-Events Model



1. Assume accidents caused by chain of failure events
2. **Identify the potential accident chains**
3. Try to prevent the identified scenarios (chains)
   a. Establish barriers between events or
   b. Prevent component failures

Forward vs. Backward Search



Failure Modes and Effects Analysis (FMEA)

Item	Failure Modes	Cause of Failure	Possible Effects	Prob.	Level	Possible actions
Motor Case	Rupture	1. Poor workmanship 2. Defective materials 3. Damage during trans. 4. Damage during handling 5. Overpressurization	Destruction of missile	0.0006	Critical	Quality control …

What accident causality model underlies this?

Failure Modes and Effects Analysis (FMEA)

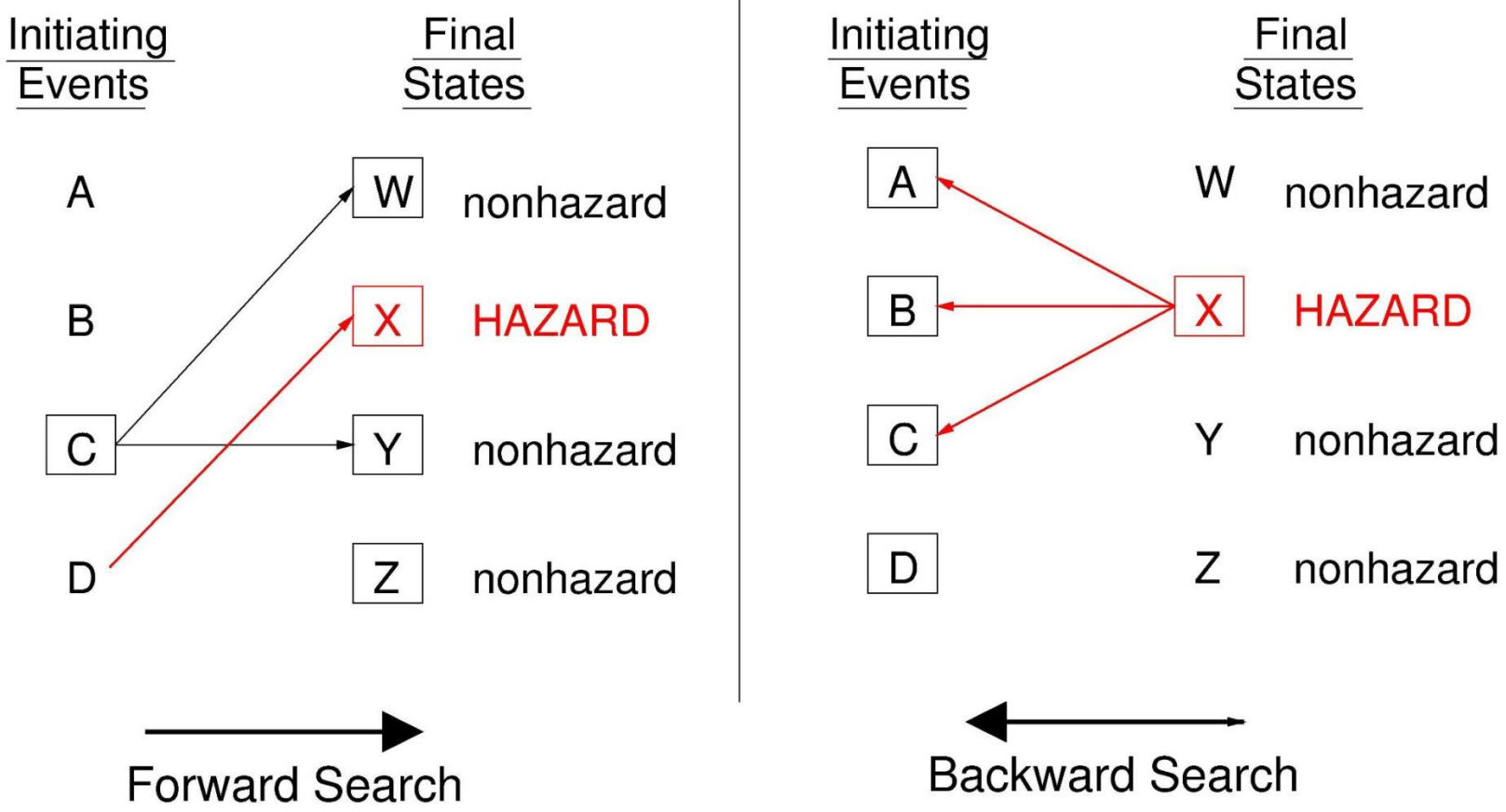
Item	Failure Modes	Cause of Failure	Possible Effects	Prob.	Level	Possible actions
Motor Case	Rupture	1. Poor workmanship 2. Defective materials 3. Damage during trans. 4. Damage during handling 5. Overpressurization	Destruction of missile	0.0006	Critical	Quality control …

What accident causality model underlies this?

Limitations

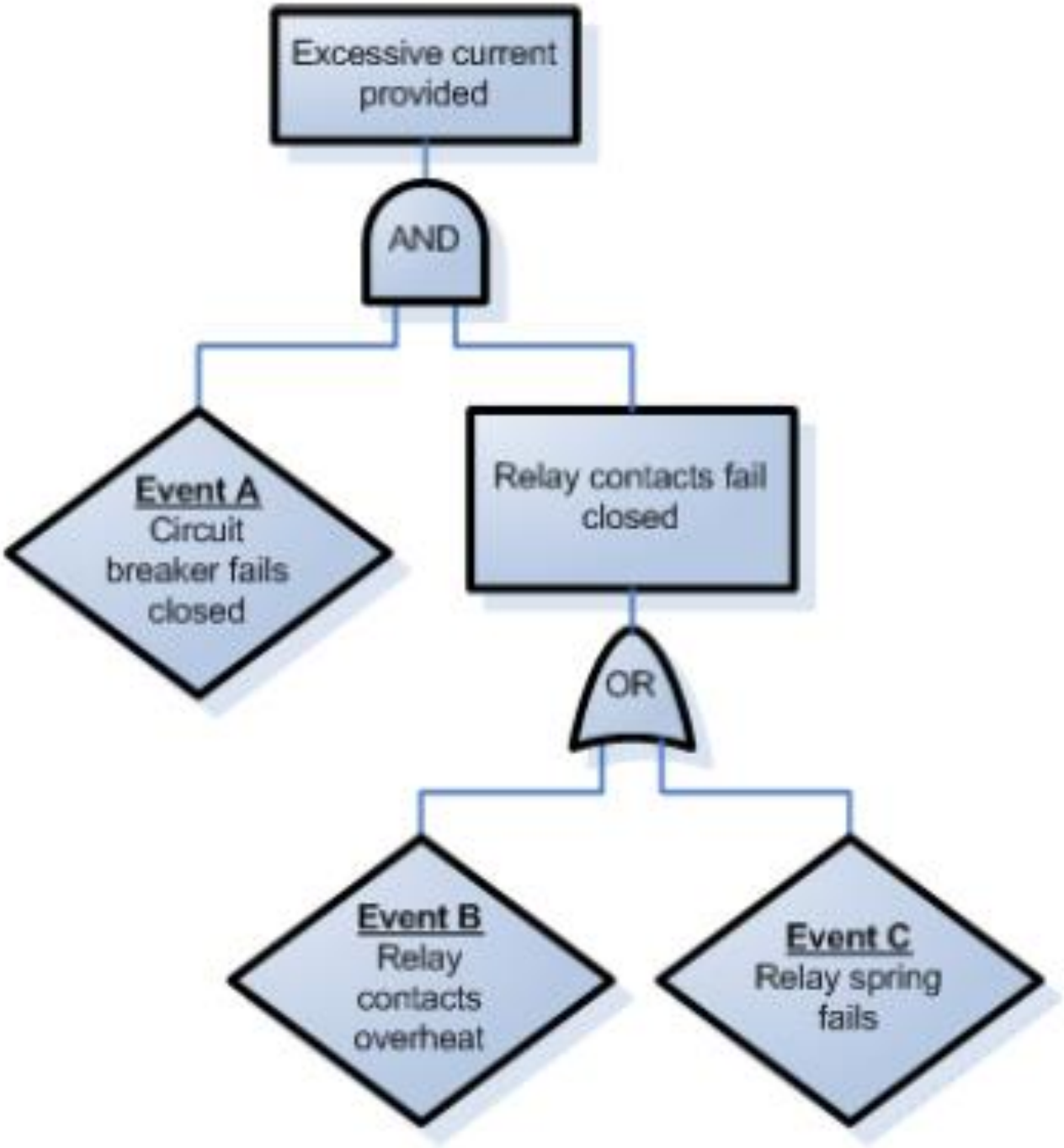
- Assumes accidents caused by system component failure
- Single component failures only
- Requires detailed system design (limits early analysis)
- Works best on hardware/mechanical components
  - Not software, human operators, organizational factors
- Inefficient, analyzes important + unimportant
  - Can result in thousands of pages of worksheets
- Tends to encourage redundancy as a solution (which may not be very efficient and may be very costly)
- Failure modes must already be known
  - Best for standard parts with few and well-known failure modes

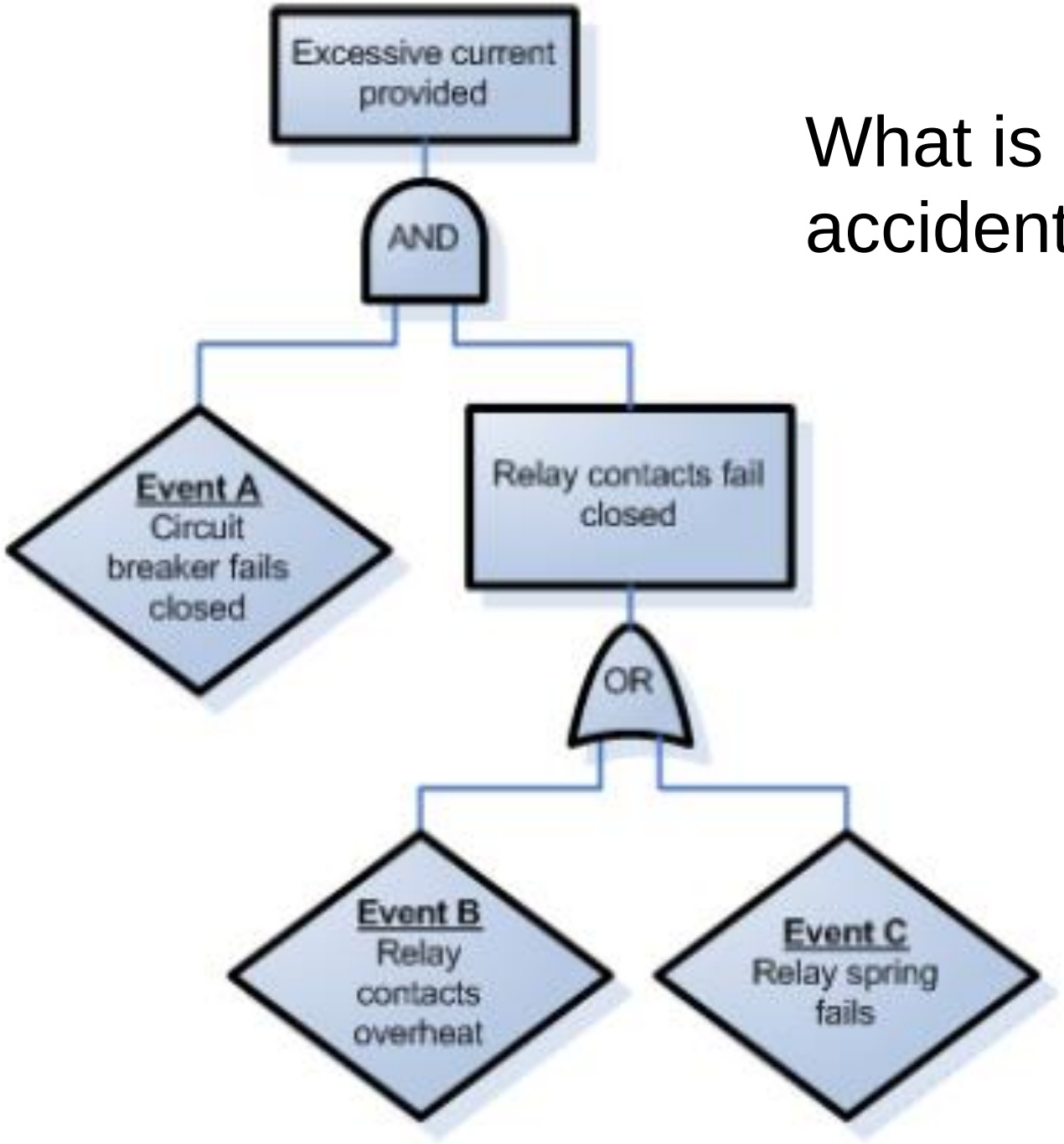
Forward vs. Backward Search



Fault Tree Analysis (FTA)

- 1961: Bell Labs and Boeing analysis of the Minuteman Missile System
- Today one of most popular hazard analysis (risk assessment) methods
- Backward search
  - Start with undesirable event at top of tree
  - Goal is to identify causal chain of failures leading to the event

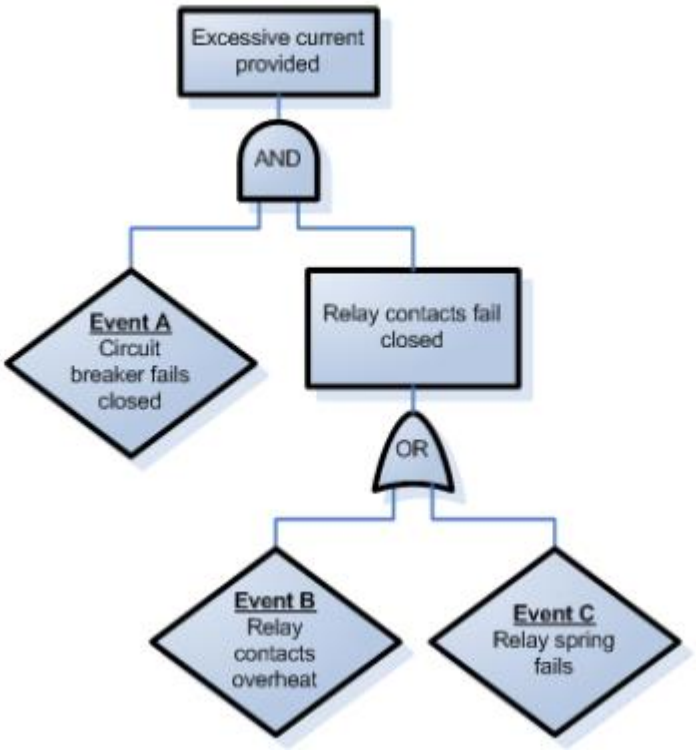

Excessive current provided
AND
Event A
Circuit breaker fails closed
Relay contacts fail closed
OR
Event B
Relay contacts overheat
Event C
Relay spring fails

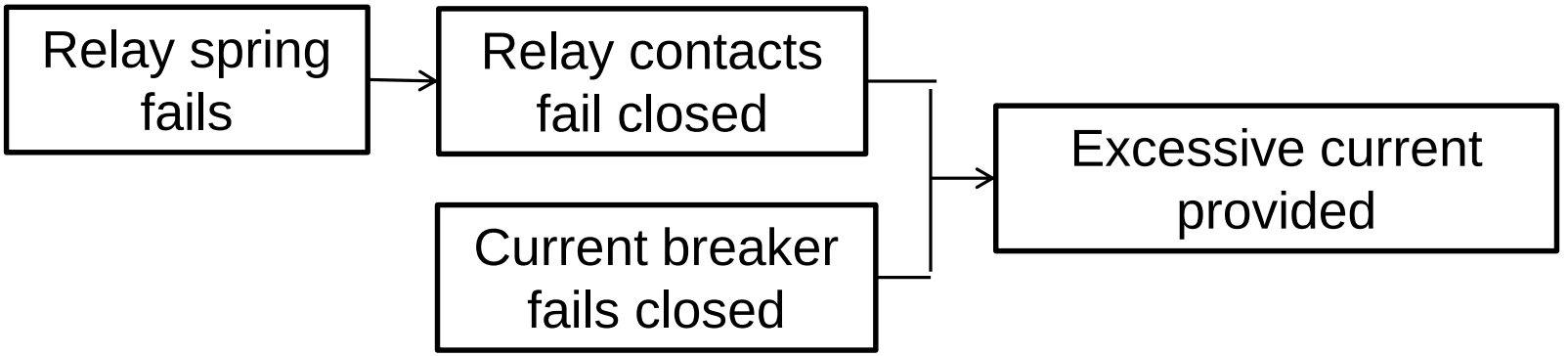


What is underlying accident model?

Chain-of-Failure Events

Fault Tree:

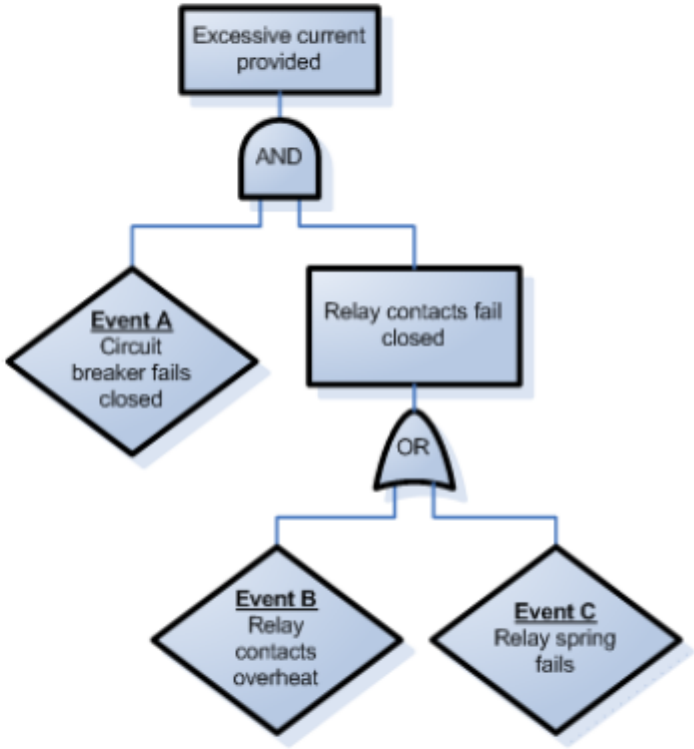




Quantitative Fault Tree Analysis

- If we can assign probabilities to lowest boxes…
  - Can propagate up using probability theory
  - Can get overall total probability of hazard!

- AND gate
  - P(A and B) = P(A) * P(B)
- OR gate
  - P(A or B) = P(A) + P(B)



Any assumptions being made?

Quantitative Fault Tree Analysis

- If we can assign probabilities to lowest boxes…
  - Can propagate up using probability theory
  - Can get overall total probability of hazard!

- AND gate
  - P(A and B) = P(A) * P(B)
- OR gate
  - P(A or B) = P(A) + P(B)

Only if events A,B are independent!

Fault Tree Exercise

- **Hazard**: Over-pressurization
- **Design**:

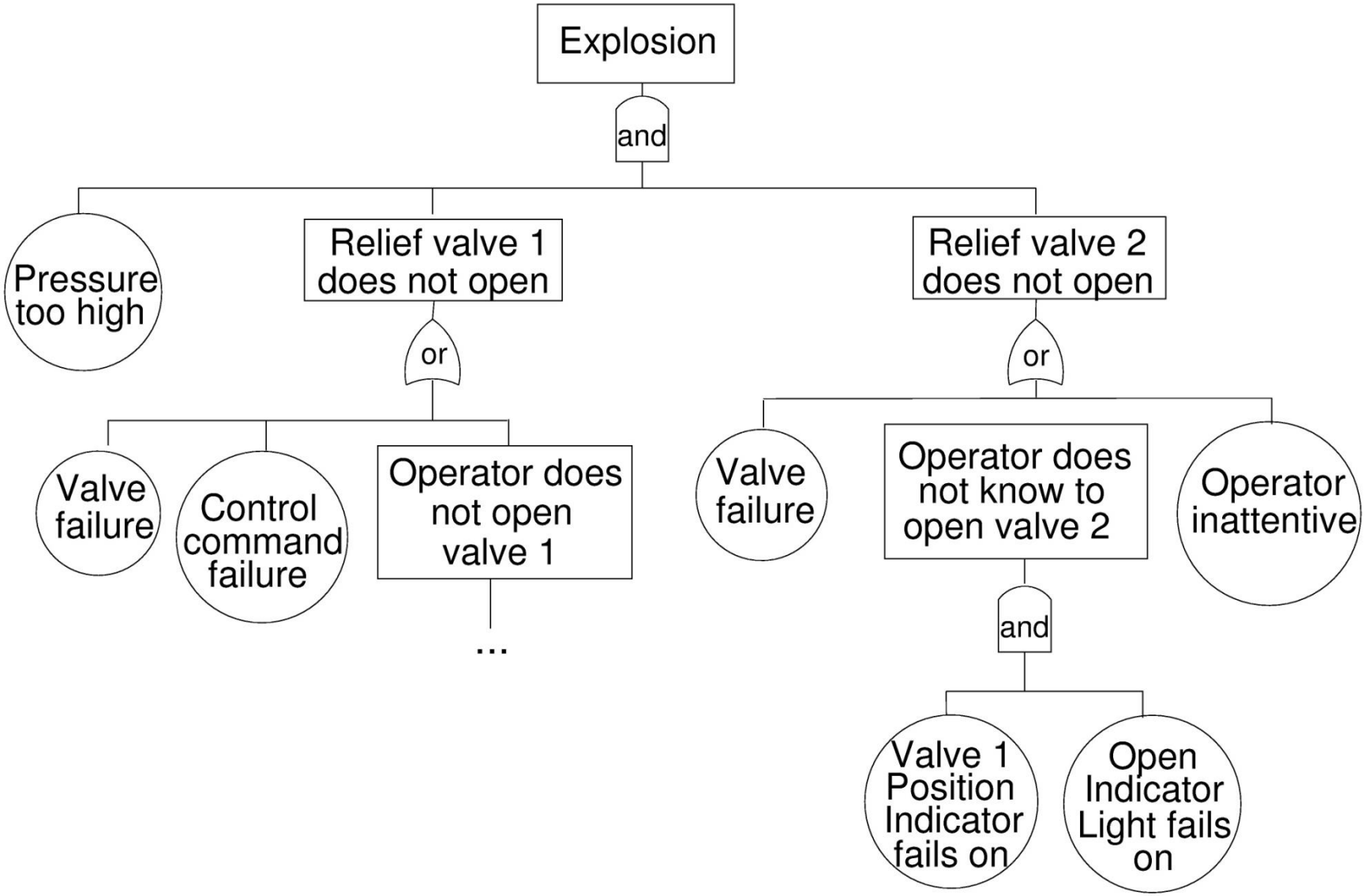
  System includes a relief valve opened by an operator to protect against over-pressurization. A secondary valve is installed as backup in case the primary valve fails. The operator must know if the primary valve does not open so the backup valve can be activated.

  Operator console contains both a primary valve position indicator and a primary valve open indicator light.

Exercise

- Build a fault tree for this system.
  - What is the top event?

Fault Tree Example



Example of Unrealistic Risk Assessment Leading to an Accident

- **Events**: The open position indicator light and open indicator light both illuminated. However, the primary valve was NOT open, and the system exploded.
- **Causal Factors**:
  - Post-accident examination discovered the indicator light circuit was wired to indicate presence of power at the valve, but it did not indicate valve position.
  - Thus, the indicator showed only that the activation button had been pushed, not that the valve had opened.
  - An extensive quantitative safety analysis of this design had assumed a low probability of simultaneous failure for the two relief valves, but ignored the possibility of design error in the electrical wiring; the probability of design error was not quantifiable.
  - No safety evaluation of the electrical wiring was made; instead, confidence was established on the basis of the low probability of coincident failure of the two relief valves.

FTA Strengths

- Captures combinations of failures
- More efficient than FMEA
  - Analyzes only failures relevant to top-level event
- Provides a graphical format to help in understanding the analysis results
- Analyst has to think about the system in great detail during tree construction
- Provides insight into weak points (e.g., common mode failures) of system designs

FTA Limitations

- Difficult to capture delays and other temporal factors
- Transitions between states of operational phases not represented
- Very labor intensive for any but simplest systems
  e.g., one Embraer IMA fault tree was 2200 pages
- No common model working from (all in analyst's head)
- Very difficult to do for anything but a very simple system
- Can become complex very quickly, difficult to review
- Looks only at simple interactions among component failures
- Quantification not realistic for anything but hardware failures [SAE ARP 4761]

Other Hazard Analysis Techniques Used

- Event tree analysis or ETA (used in process industry)
  - Start with failure event (e.g., overheating of fuel) and work forward (how respond to event to prevent losses)
- HAZOP (HAZard and OPerability) analysis (chemical and process industry)
- HFACS (Human Factors Accident and Classification System)
  - Based on Reason's Swiss Cheese Model
  - Looks only for human errors
  - Used to investigate the cause of accidents (after the fact)

Limitations of Traditional Analysis Methods

- Assumption of independence
- Simple chain-of-failure events model
- Cannot handle complex human behavior
- Cannot handle
  - Software (system design and requirements flaws)
  - Non-discrete (continuous) events
  - Timing problems
  - Systemic factors (e.g., managerial/production pressures)
  - Complex systems
- Cannot handle accidents involving non-failures

More Limitations

- Component failure accidents only
  - Not accidents arising from interactions among non-failed components, e.g., system design flaws
- Single component failures only
  - What about non-events (systemic factors)? safety culture?, conditions that influence behavior, changes over time …
- Requires detailed system design (limits early analysis)
- Works best on hardware/mechanical components
  - Not software, human operators, organizational factors

Current State of the Art: PRA

- Risk and Risk Assessment
  - Little data validating PRA or methods for calculating it
  - Other problems
    - May be significant divergence between modeled system and as-built and as-operated system
    - Interactions between social and technical part of system may invalidate technical assumptions underlying analysis
    - Effectiveness of mitigation measures may change over time
  - Why are likelihood estimates inaccurate in practice?
    - Important factors left out (operator error, flawed decision making, software) because don't have probability estimates
    - Non-stochastic factors involved in events
    - Heuristic biases

Heuristic Biases

- Confirmation bias (tend to deny uncertainty and vulnerability)
  - People look for evidence that supports their hypothesis
  - Reject evidence that does not
- Construct simple causal scenarios
  - If none comes to mind, assume event is impossible
- Tend to identify simple, dramatic events rather than events that are chronic or cumulative
- Incomplete search for causes
  - Once one cause identified and not compelling, then stop search
- Defensive avoidance
  - Downgrade accuracy or don't take seriously
  - Avoid topic that is stressful or conflicts with other goals

Controlling Heuristic Biases

- Cannot eliminate completely but can reduce
- Use structured method for assessing and managing “risk”
  - Following a structured process and rules to follow can diminish power of biases and encourage more thorough search
  - Concentrate on causal mechanisms vs. likelihood
- Use worst case analysis (vs. “design basis accident”)
- “Prove” unsafe rather than “safe”
  - Hazard analysis vs. safety case

Boeing 787 Lithium Battery Fires

Models predicted 787 battery thermal problems would occur once in 10 million flight hours…but two batteries overheated in just two weeks in 2013

Boeing 787 Lithium Battery Fires

- A module monitors for smoke in the battery bay, controls fans and ducts to exhaust smoke overboard.
- Power unit monitors for low battery voltage, shut down various electronics, including ventilation
- Smoke could not be redirected outside cabin

All software requirements were satisfied!
The requirements were unsafe

Why do we need something new?

Our current tools are all 40-65 years old but our technology is very different today

1940 1950 1960 1970 1980 1990 2000 2010 2020

FMEA

FTA HAZOP

ETA

Bow Tie (CCA) FTA + ETA

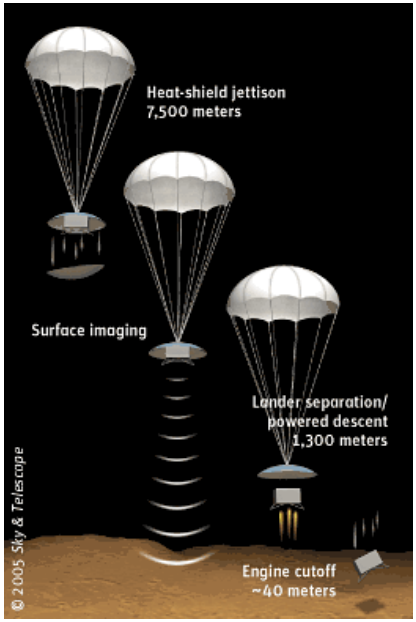
- Introduction of computer control
- Exponential increases in complexity
- New technology
- Changes in human roles

Assumes accidents caused by component failures

What Failed Here?

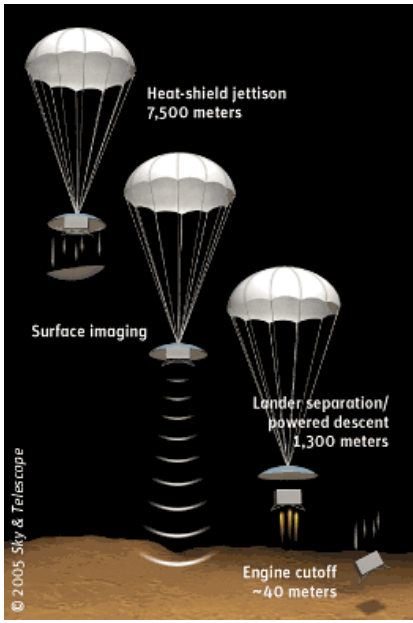
- Navy aircraft were ferrying missiles from one location to another.
- One pilot executed a planned test by aiming at aircraft in front and firing a dummy missile.
- Nobody involved knew that the software was designed to substitute a different missile if the one that was commanded to be fired was not in a good position.
- In this case, there was an antenna between the dummy missile and the target so the software decided to fire a live missile located in a different (better) position instead.

Accident with No Component Failures



- Mars Polar Lander
  - Have to slow down spacecraft to land safely
  - Use Martian atmosphere, parachute, descent engines (controlled by software)
  - Software knows landed because of sensitive sensors on landing legs. Cuts off engines when determines have landed.
  - But “noise” (false signals) by sensors generated when landing legs extended. Not in software requirements.
  - Software not supposed to be operating at that time but software engineers decided to start early to even out the load on processor
  - Software thought spacecraft had landed and shut down descent engines while still 40 meters above surface

Accident with No Component Failures



- Mars Polar Lander
  - Have to slow down spacecraft to land safely
  - Use Martian atmosphere, parachute, descent engines (controlled by software)
  - Software knows landed because of sensitive sensors on landing legs. Cuts off engines when determines have landed.
  - But "noise" (false signals) by sensors generated when landing legs extended. Not in software requirements.
  - Software not supposed to be operating at that time but software engineers decided to start early to even out the load on processor
  - [illegible]nt

All software requirements were satisfied!
The requirements were unsafe

Washington State Ferry Problem

- Rental cars could not be driven off ferries when got to port
- Local rental car company installed a security device to prevent theft by disabling cars if car moved when engine stopped
- When ferry moved and cars not running, disabled them.

Warsaw A320 Accident

- Software protects against activating thrust reversers when airborne
- Hydroplaning and other factors made the software not think the plane had landed
- Pilots could not activate the thrust reversers and ran off end of runway into a small hill.

Toyota Unintended Acceleration

- **2004:** Push-button ignition
- **2004-2009**
  - 102 incidents of uncontrolled acceleration
  - Speeds exceed 100 mph despite stomping on the brake
  - 30 crashes
  - 20 injuries
- **Today**
  - Software fixes for pushbutton ignition, pedals

Pushbutton was reliable!
Software was reliable!

http://www.reuters.com/article/2010/07/14/us-toyota-idUSTRE66D0FR20100714
http://www.statesman.com/business/u-s-toyota-cite-driver-error-in-many-803504.html

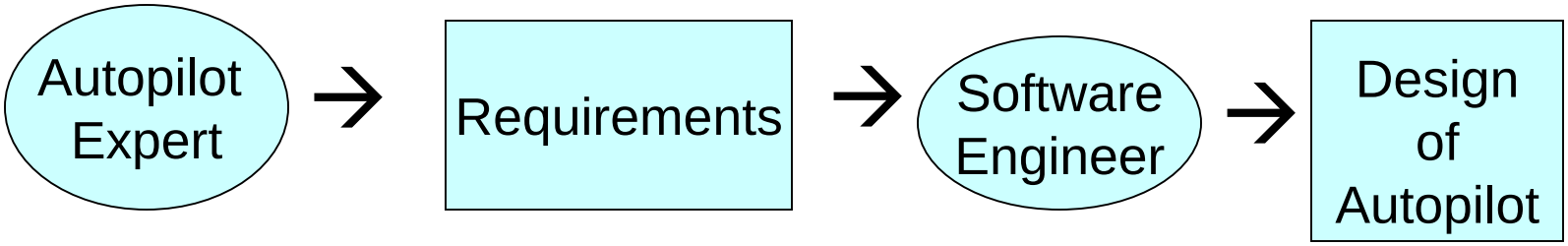
Toyota

- **2004:** Push-button ignition
- **2004-2009**
  - 102 incidents of uncontrolled acceleration
  - Speeds exceed 100 mph despite stomping on the brake
  - 30 crashes
  - 20 injuries
- **2009, Aug**:
  - Car accelerates to 120 mph
  - Passenger calls 911, reports stuck accelerator
  - Car crashes killing 4 people
  - Driver was offensive driving instructor for police
- **Today**
  - Software fixes for pushbutton ignition, pedals

All component requirements were met…
Yet system behavior was unexpected, unsafe!

http://www.reuters.com/article/2010/07/14/us-toyota-idUSTRE66D0FR20100714
http://www.statesman.com/business/u-s-toyota-cite-driver-error-in-many-803504.html

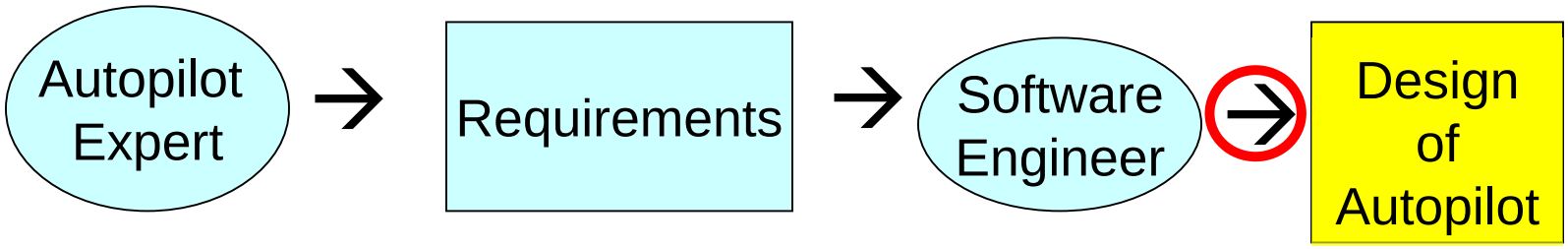
What is Software?



- Software is design abstracted from its physical realization
- Designs don't "fail" and they do not behave randomly (are not stochastic)

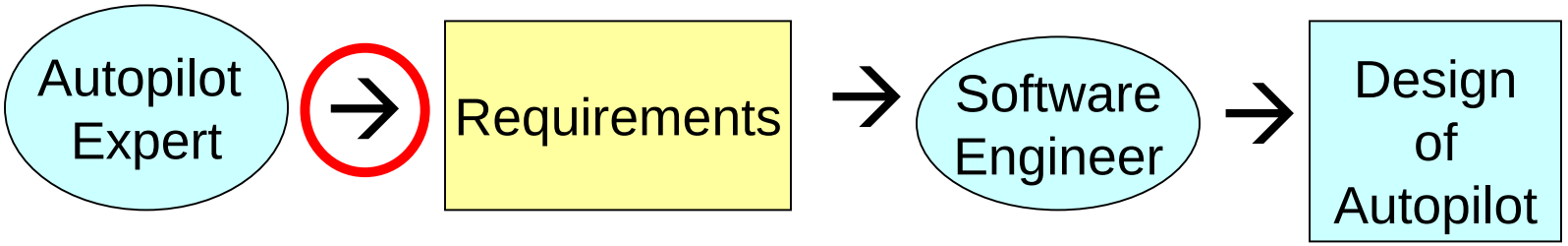
The role of software in accidents almost always involves flawed requirements

- Incomplete or wrong assumptions about operation of controlled system or required operation of computer
- Unhandled controlled-system states and environmental conditions



The role of software in accidents almost always involves flawed requirements

- Incomplete or wrong assumptions about operation of controlled system or required operation of computer
- Unhandled controlled-system states and environmental conditions

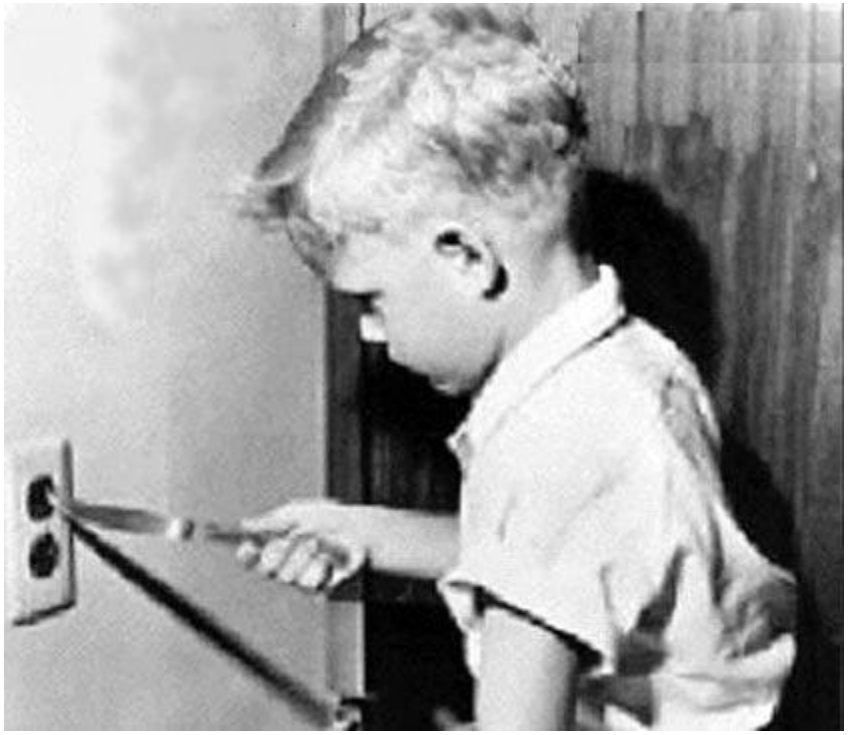


Only trying to get the software “correct” or to make it reliable will not make it safer under these conditions

Context is Important

Safe or Unsafe?

Safety Depends on Context



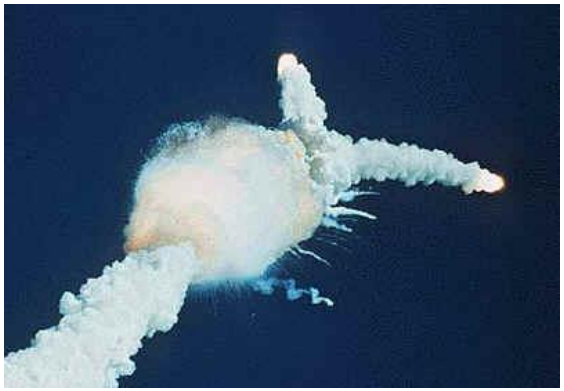
Individual components not inherently safe or unsafe

Safety is a System Property

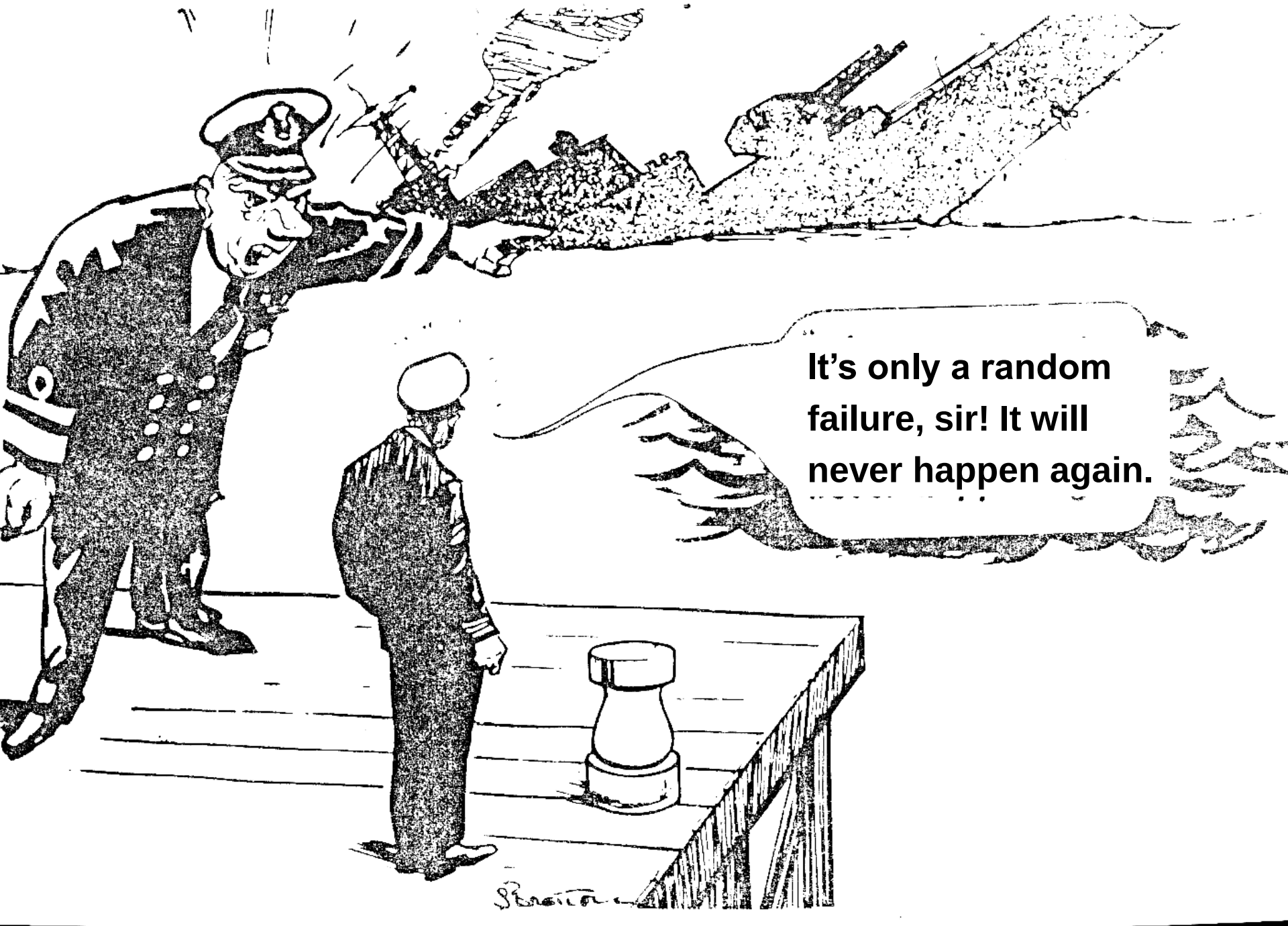
- Context determines whether something is safe or not

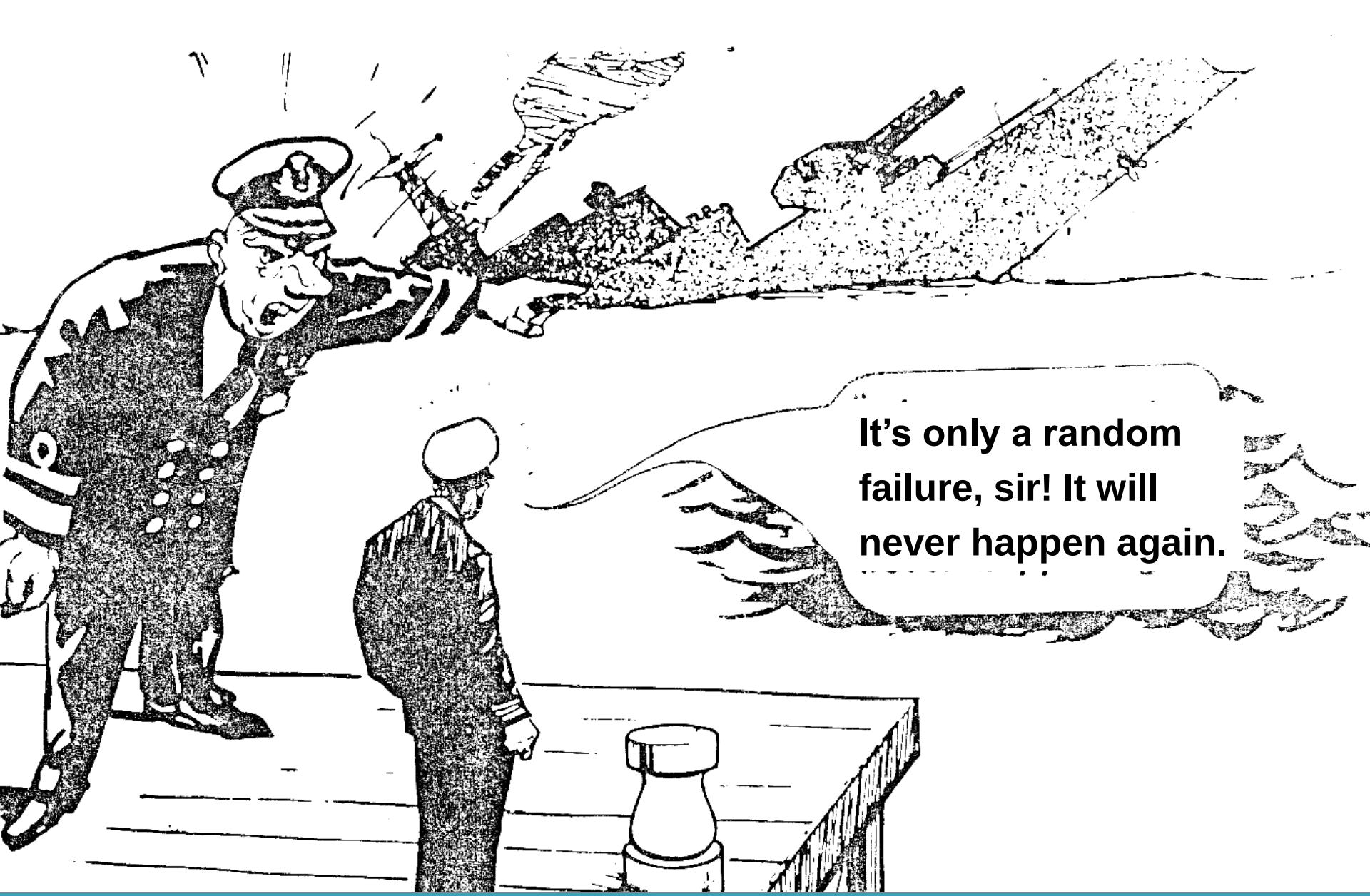
Ariane 4

Ariane 5



- Used same inertial reference software
- Especially true for software and human behavior


It's only a random failure, sir! It will never happen again.

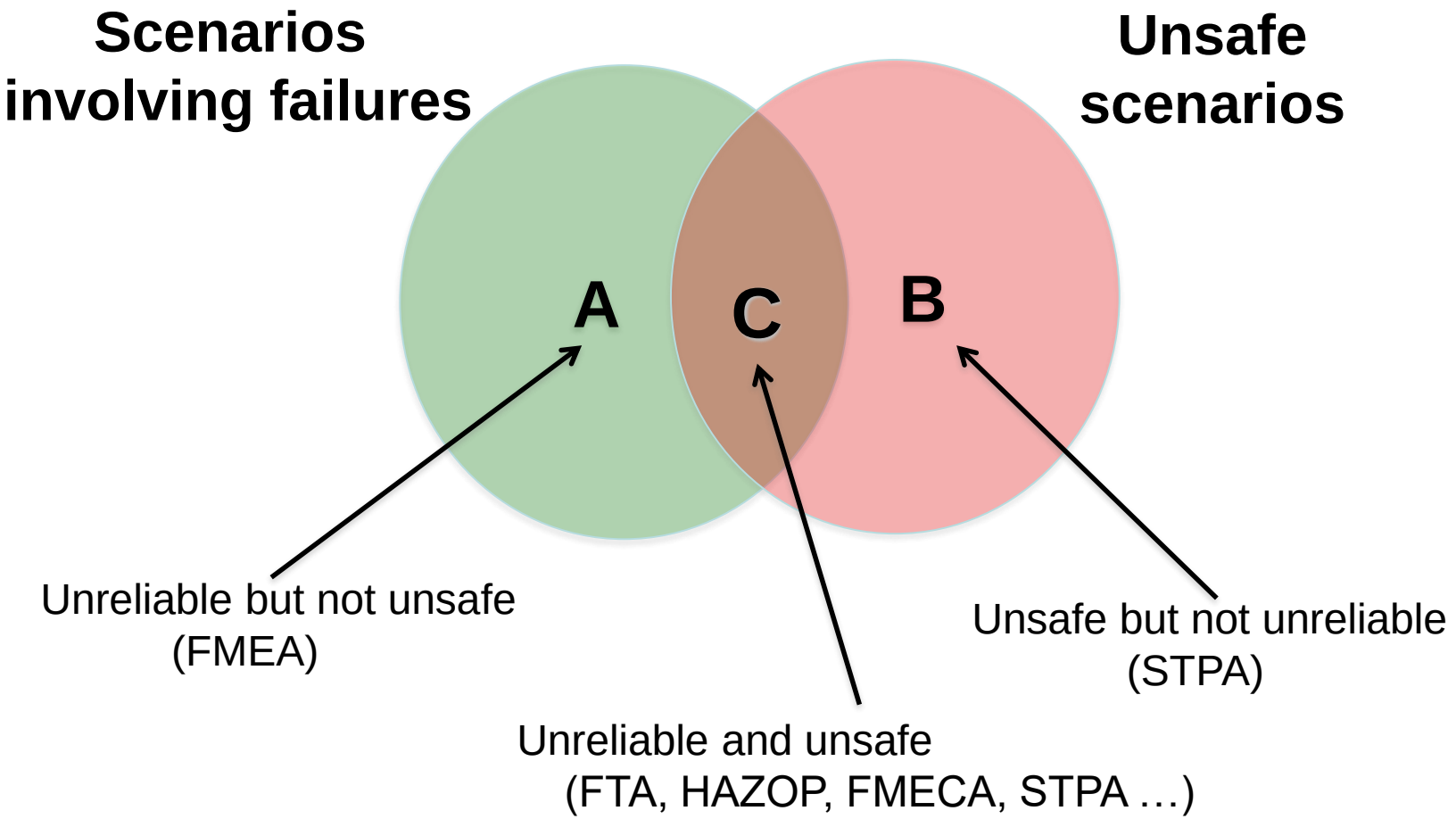


Reliability and safety are different properties today

Two Types of Accidents

- **Component Failure Accidents**
  - Single or multiple component failures
  - Usually assume random failure
- **Component Interaction Accidents**
  - Arise in interactions among components
  - Related to complexity (coupling) in our system designs, which leads to system design and system engineering errors
  - No components may have “failed”
  - Exacerbated by introduction of computers and software but the problem is system design errors

Confusing Safety and Reliability



Preventing Component or Functional Failures is Not Enough

Another Accident Involving Thrust Reversers

- Tu-204, Moscow, 2012
- Red Wings Airlines Flight 9268
- The soft 1.12g touchdown made runway contact a little later than usual.
- With the crosswind, this meant weight-on-wheels switches did not activate and the thrust-reverse system would not deploy.

An Accident Involving Thrust Reversers (2)

- Pilots believe the thrust reversers are deploying like they always do. With the limited runway space, they quickly engage high engine power to stop quicker. Instead this accelerated the Tu-204 forwards, eventually colliding with a highway embankment.

An Accident Involving Thrust Reversers (2)

- Pilots believe the thrust reversers are deploying like they always do. With the limited runway space, they quickly engage high engine power to stop quicker. Instead this accelerated the Tu-204 forwards, eventually colliding with a highway embankment.

In complex systems, human and technical considerations cannot be isolated

Human factors concentrates on the “screen out”

Hardware/software engineering concentrates on the “screen in”

Not enough attention on integrated system as a whole

(e.g, mode confusion, situation awareness errors, inconsistent behavior, etc.

We Need Something New

- New levels of complexity, software, human factors do not fit into a reliability-oriented world.
- Two approaches being taken now:

Pretend there is no problem

Shoehorn new technology and new levels of complexity into old methods

Summary of the Problem:

- We need models and tools that handle:
  - Hardware and hardware failures
  - Software (particularly requirements)
  - Human factors
  - Interactions among system components
  - System design errors
  - Management, regulation, policy
  - Environmental factors
  - Intrusions (security problems)
  - “Migration toward higher risk” (changes over time)

And the interactions among all these things

Paradigm Change

- Does not imply what previously done is wrong and new approach correct
- Einstein:

  "Progress in science (moving from one paradigm to another) is like climbing a mountain"

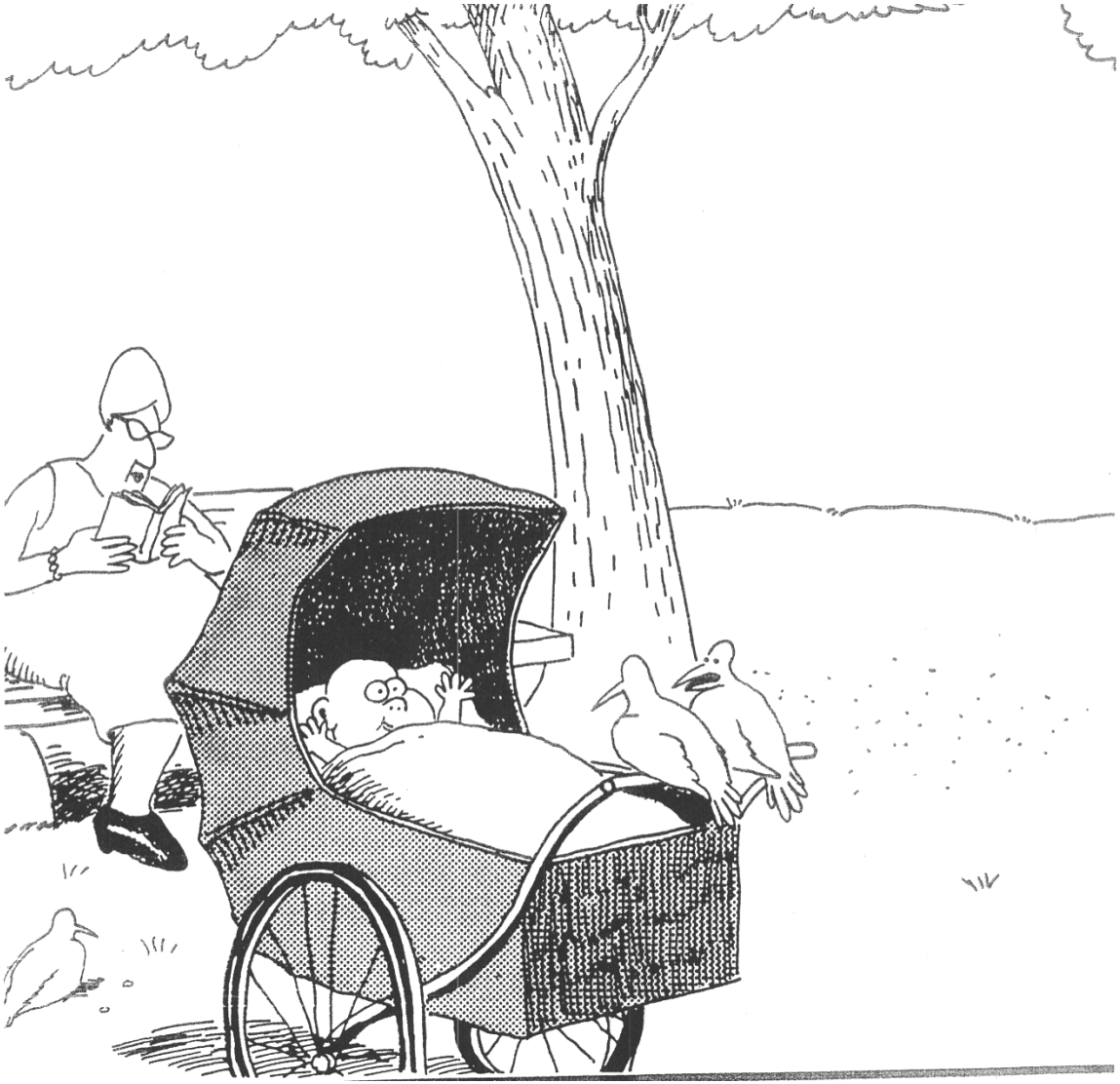
As move further up, can see farther than on lower points

Paradigm Change (2)

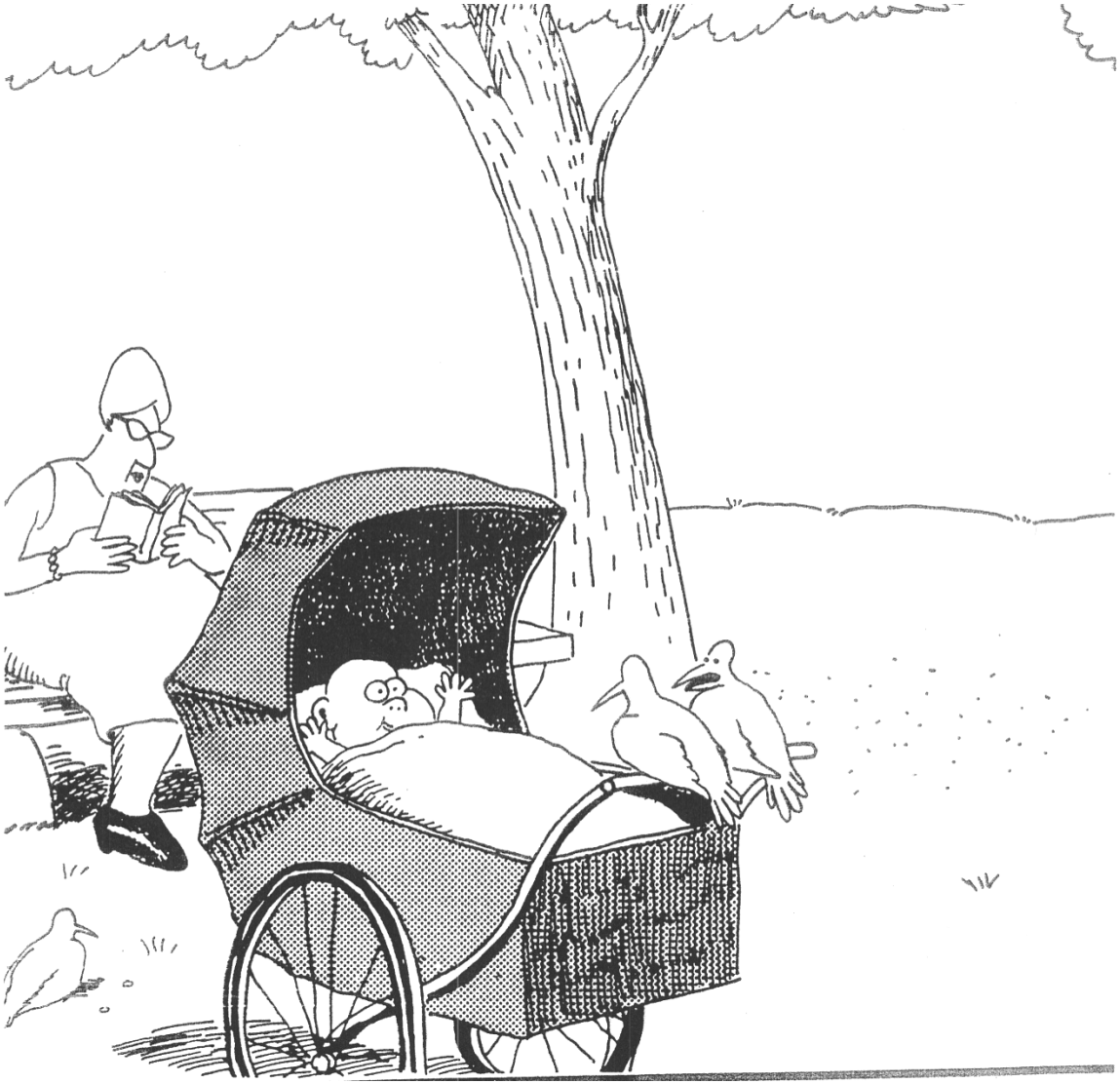
New perspective does not invalidate the old one, but extends and enriches our appreciation of the valleys below

Value of new paradigm often depends on ability to accommodate successes and empirical observations made in old paradigm.

New paradigms offer a broader, rich perspective for interpreting previous answers.



It’s still hungry … and I’ve been stuffing worms into it all day.



It’s still hungry … and I’ve been stuffing worms into it all day.

We Need New Tools for the New Problems

A new systems-theoretic approach to safety engineering

The Problem is Complexity

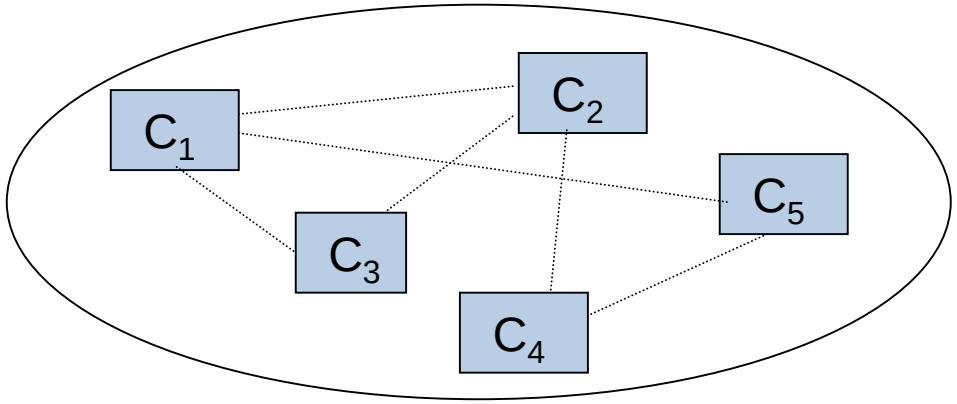
Ways to Cope with Complexity

- Analytic Decomposition
- Statistics
- Systems Theory

Analytic Decomposition ("Divide and Conquer")

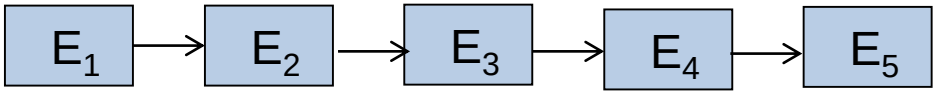
1. Divide system into separate parts

Physical/Functional: Separate into distinct components



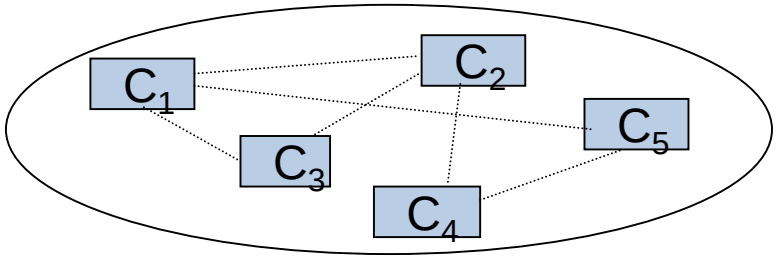
Components interact
In direct ways

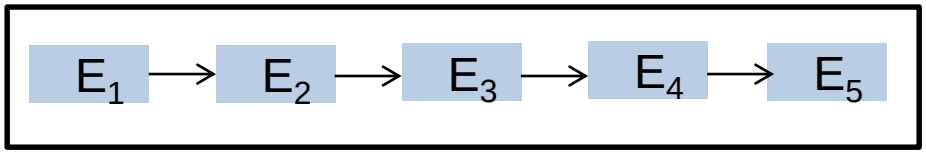
Behavior: Separate into events over time



Each event is the direct
result of the preceding event

Analytic Decomposition (2)





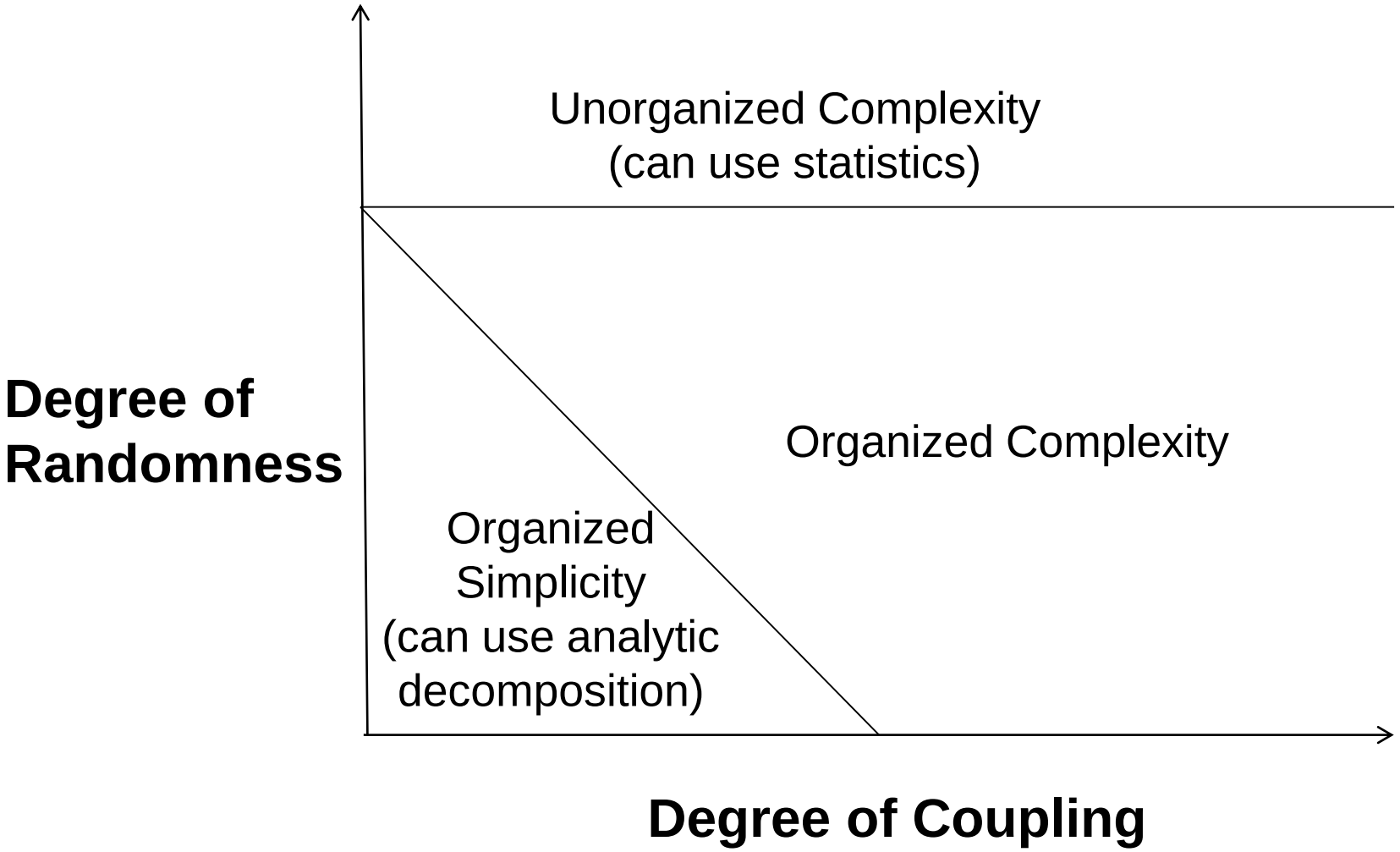
2. Analyze/examine pieces separately and combine results

- Assumes such separation does not distort phenomenon
  - ✓ Each component or subsystem operates independently
  - ✓ Components act the same when examined singly as when playing their part in the whole
  - ✓ Components/events not subject to feedback loops and non-linear interactions
  - ✓ Interactions can be examined pairwise

Bottom Line

- These assumptions are no longer true in our
  - Tightly coupled
  - Software intensive
  - Highly automated
  - Connected

  engineered systems
- Need a new theoretical basis
  - *System theory* can provide it



[Credit to Gerald Weinberg]

Here comes the paradigm change!

Systems Theory

- Developed for systems that are
  - Too complex for complete analysis
    - Separation into (interacting) subsystems distorts the results
    - The most important properties are emergent
  - Too organized for statistics
    - Too much underlying structure that distorts the statistics
    - New technology and designs have no historical information
- First used on ICBM systems of 1950s/1960s

System Theory was created to provide a more powerful way to deal with complexity

Systems Theory (2)

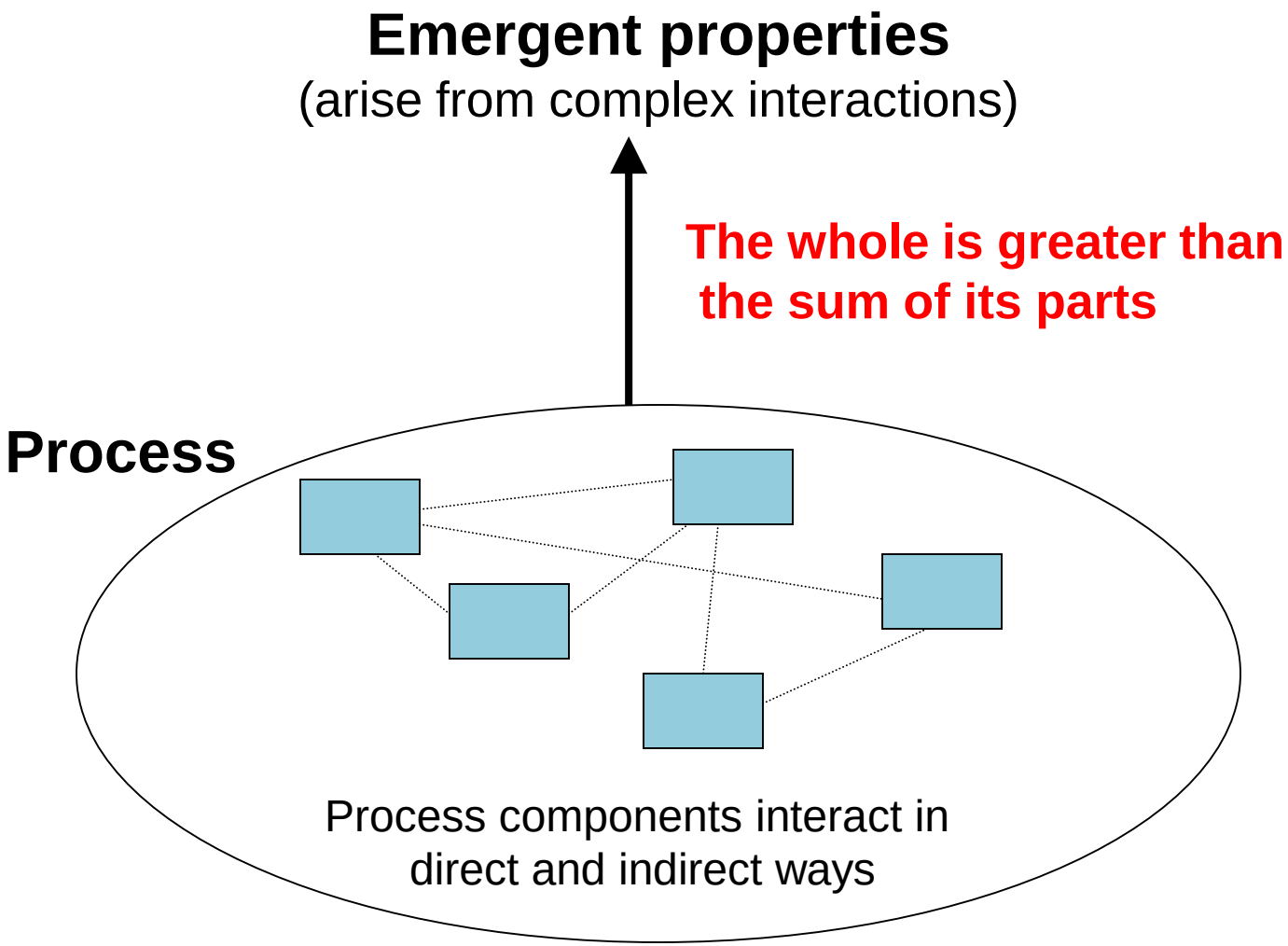
- Focuses on systems taken as a whole, not on parts taken separately
- Emergent properties
  - Some properties can only be treated adequately in their entirety, taking into account all social and technical aspects

    "The whole is greater than the sum of the parts"

  - These properties arise from relationships among the parts of the system

    How they interact and fit together

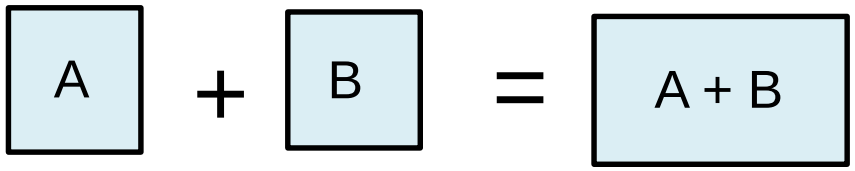
System Theory



Safety and security are emergent properties

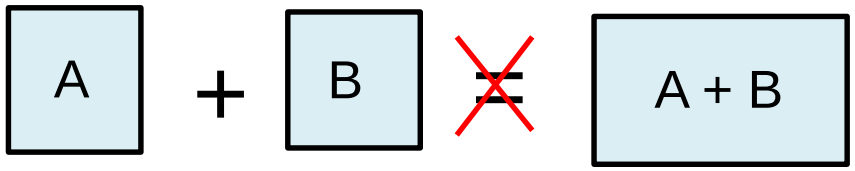
Cannot simply compose systems into a “system of systems”

- Assumption



Cannot simply compose systems into a “system of systems”

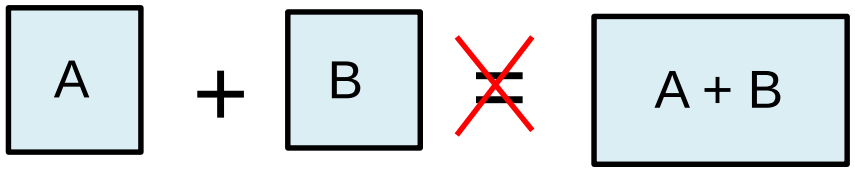
- Assumption



but **not true**

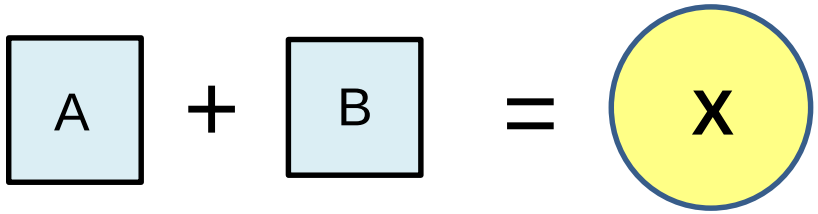
Cannot simply compose systems into a “system of systems”

- Assumption

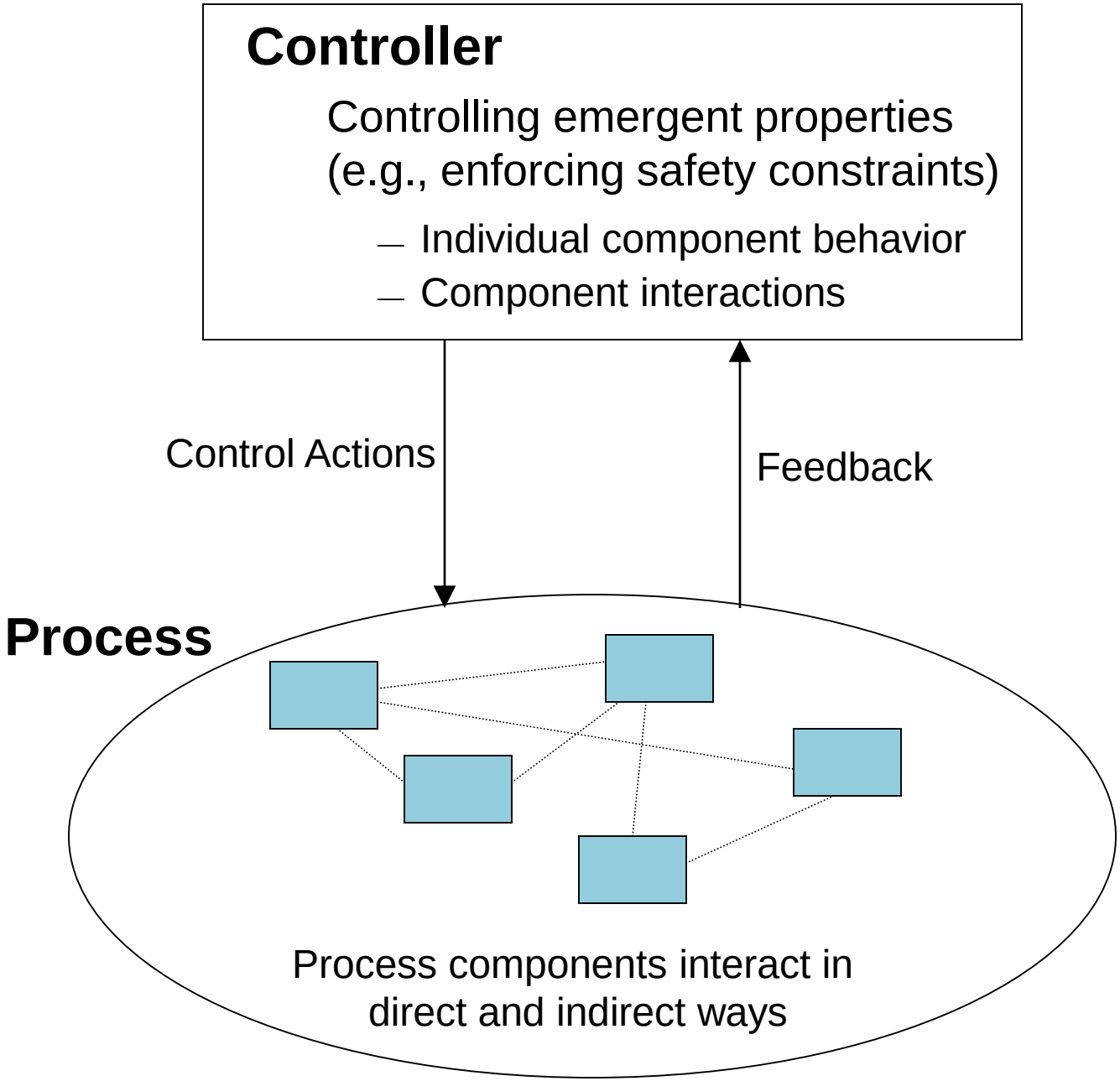


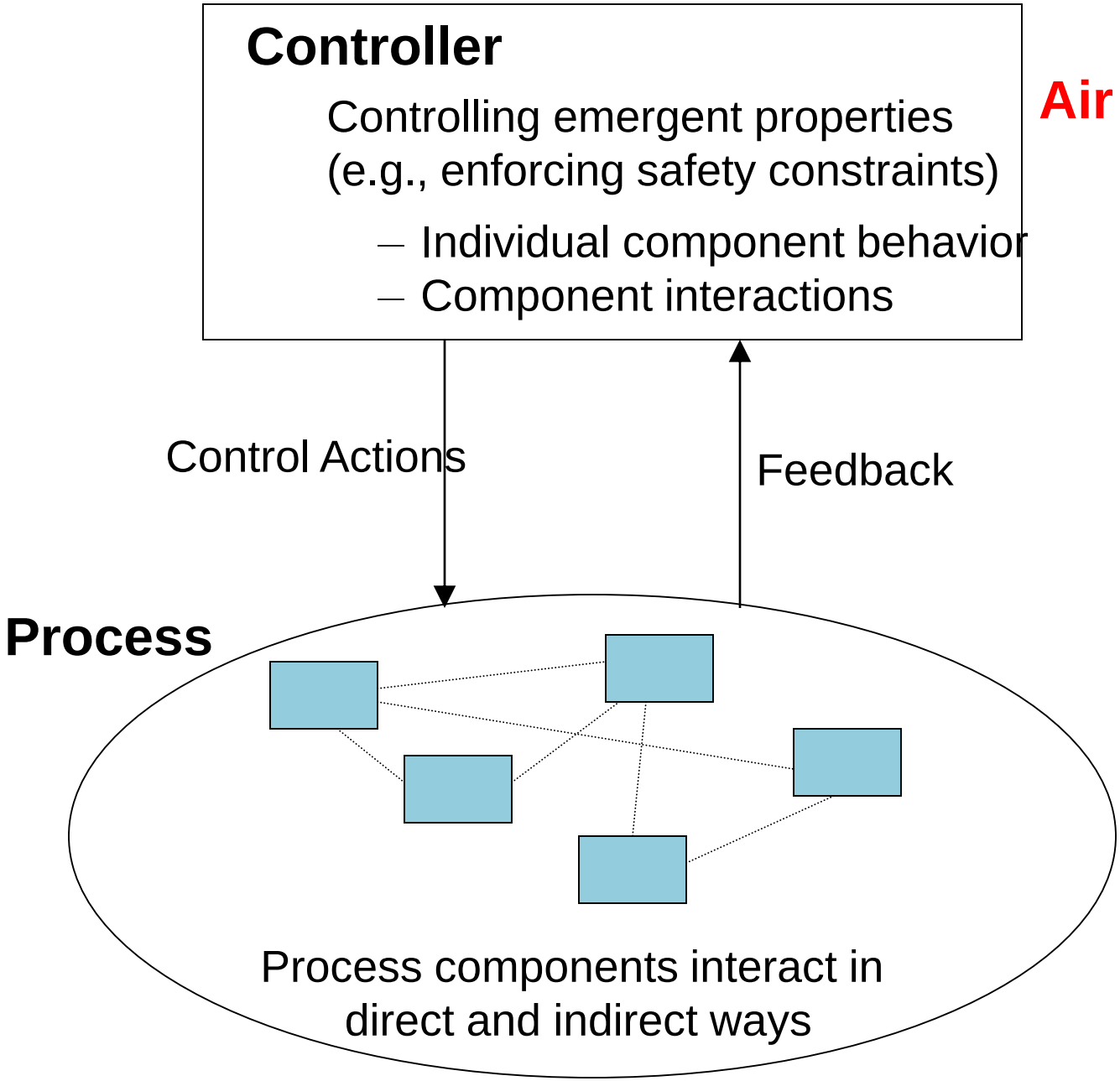
but **not true**

- In reality



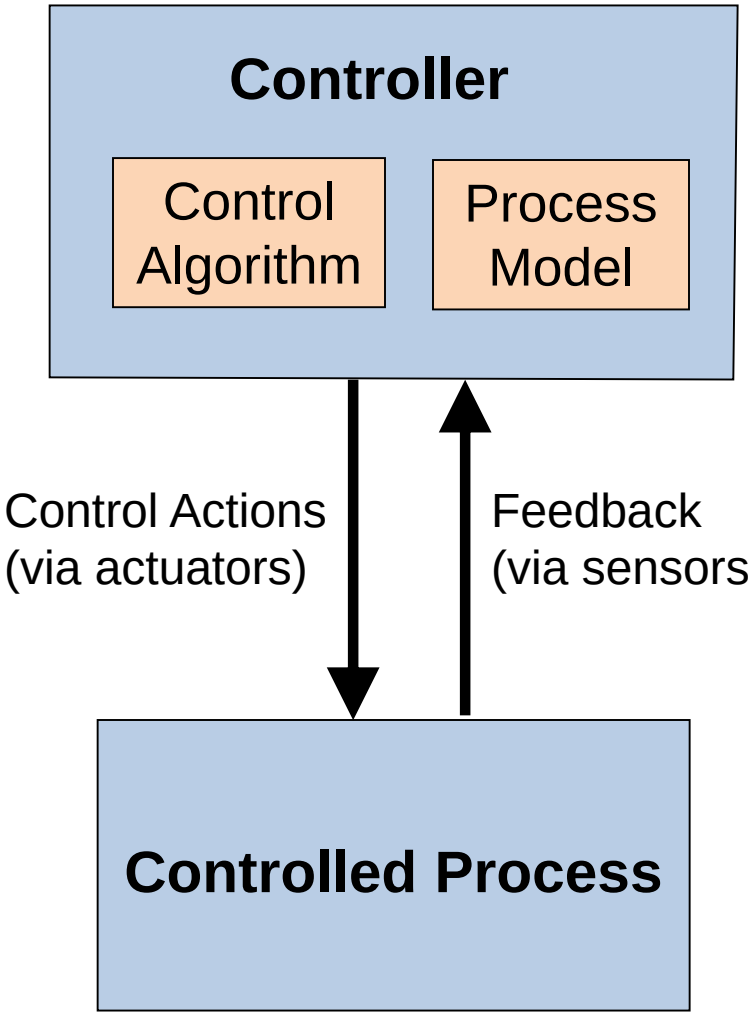
Putting two systems together gives you a new and different system with different emergent properties


Controller
Controlling emergent properties
(e.g., enforcing safety constraints)
– Individual component behavior
– Component interactions
Control Actions
Feedback
Process
Process components interact in
direct and indirect ways



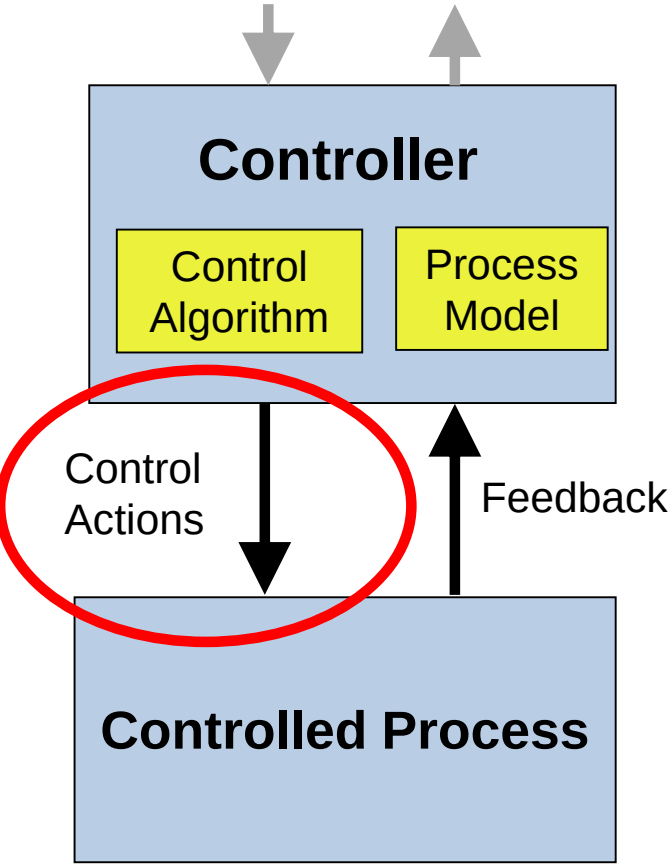
Air Traffic Control:
Safety
Throughput

Role of Process Models in Control



- Controllers use a **process model** to determine control actions
- Software/human related accidents often occur when the process model is incorrect
- Captures software errors, human errors, flawed requirements …

Unsafe Control Actions



Four types of unsafe control actions

1) Control commands required for safety are not given
2) Unsafe commands are given
3) Potentially safe commands but given too early, too late
4) Control action stops too soon or applied too long (continuous control)

Analysis:

1. Identify potential unsafe control actions
2. Identify why they might be given
3. If safe ones provided, then why not followed?

Integrated Approach to Safety and Security

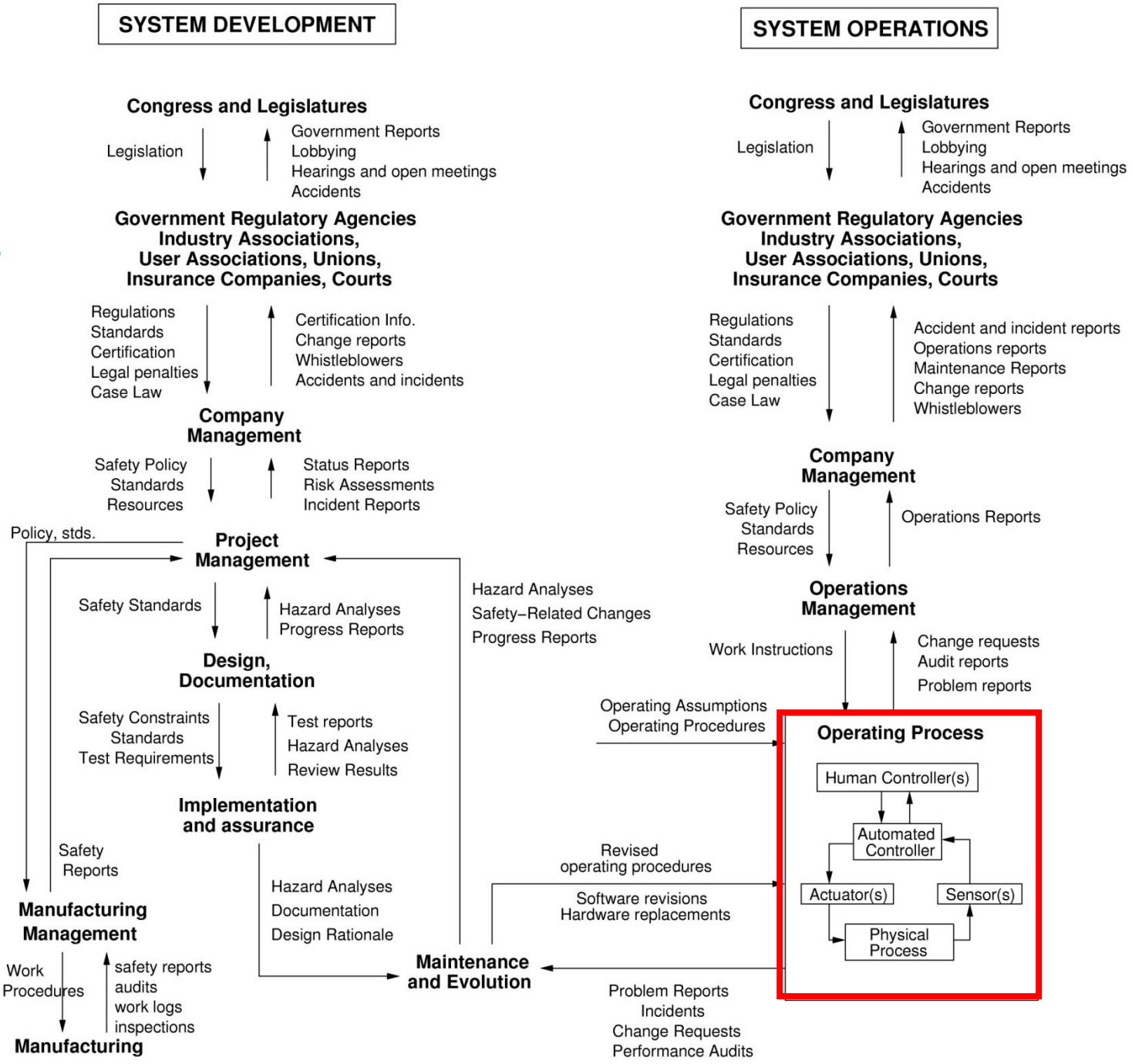
- Both concerned with losses (intentional or unintentional)
  - Ensure that critical functions and services are maintained
  - New paradigm for safety will work for security too
    - May have to add new causes, but rest of process is the same
  - A top-down, system engineering approach to designing safety and security into systems
- Paradigm change
  - Currently focus on keeping intruders out and on information security
  - Instead focus on preventing intruders from doing anything harmful if they get in and on mission assurance

Example: Stuxnet

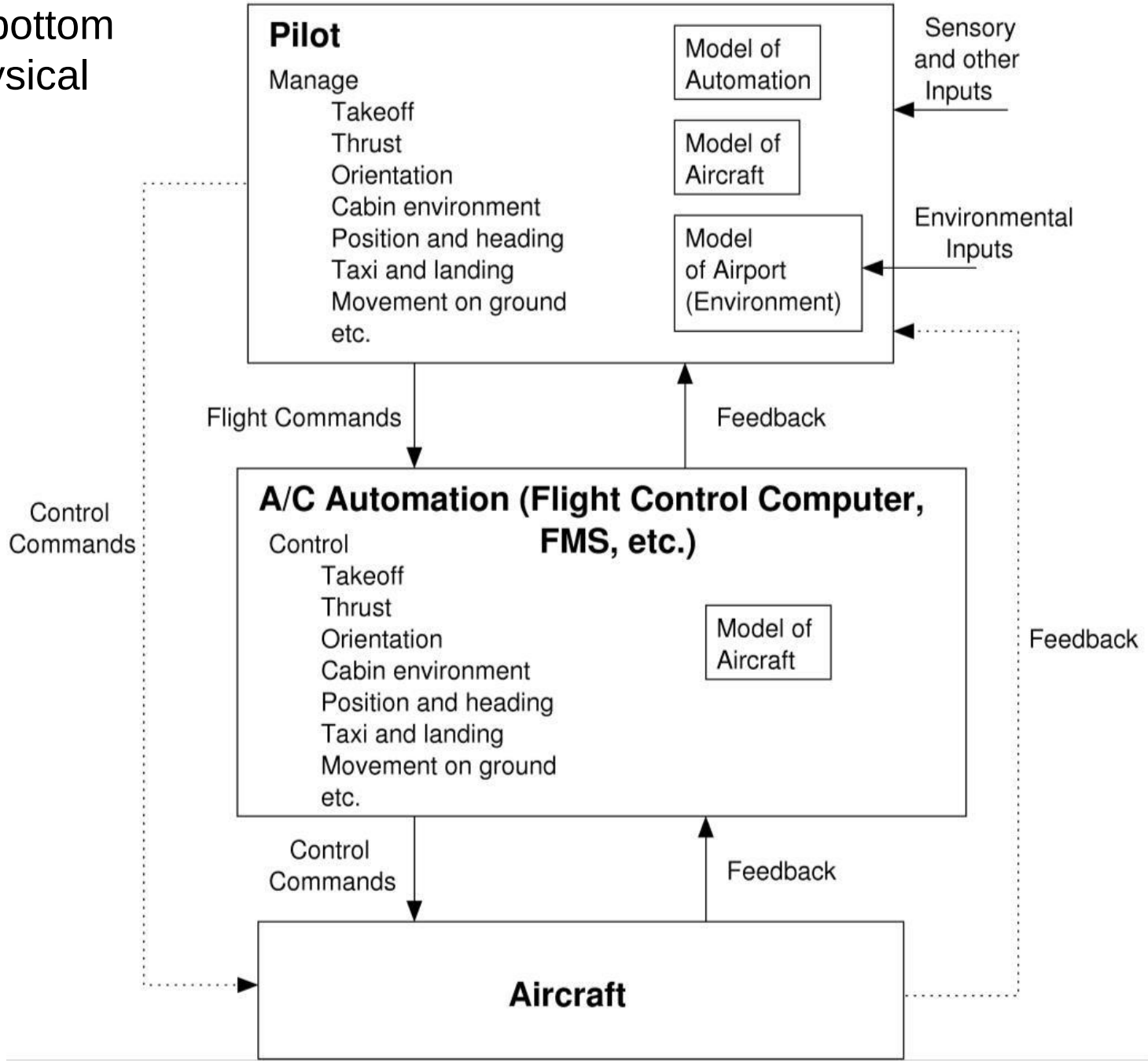
- Loss: Damage to reactor (in this case centrifuges)
- Hazard/Vulnerability: Centrifuges are damaged by spinning too fast
- Constraint to be Enforced: Centrifuges must never spin above maximum speed
- Hazardous control action: Issuing *increase speed* command when already spinning at maximum speed
- One potential causal scenario:
  - Incorrect process model: thinks spinning at less than maximum speed
    - Could be inadvertent or deliberate
- Potential controls:
  - Mechanical limiters (interlock), Analog RPM gauge

Focus on preventing hazardous state (not keeping intruders out)

Example Safety Control Structure (SMS)



[Box on bottom right, physical process]



Controls/Controllers Enforce Safety Constraints

- Power must never be on when access door open
- Two aircraft/automobiles must not violate minimum separation
- Aircraft must maintain sufficient lift to remain airborne
- Integrity of hull must be maintained on a submarine
- Toxic chemicals/radiation must not be released from plant
- Workers must not be exposed to workplace hazards
- Public health system must prevent exposure of public to contaminated water and food products
- Pressure in a offshore well must be controlled

These are the High-Level Functional Safety/Security Requirements to Address During Design

A Broad View of "Control"

Component failures and unsafe interactions may be "controlled" through design

(e.g., redundancy, interlocks, fail-safe design)

or through process

- Manufacturing processes and procedures
- Maintenance processes
- Operations

or through social controls

- Governmental or regulatory
- Culture
- Insurance
- Law and the courts
- Individual self-interest (incentive structure)

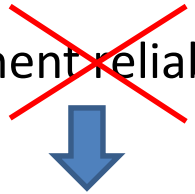
STAMP

(System-Theoretic Accident Model and Processes)

- A new, more powerful accident/loss causality model
- Based on systems theory, not reliability theory
- Defines accidents/losses as a dynamic control problem (vs. a failure problem)
- Applies to VERY complex systems
- Includes
  - Scenarios from traditional hazard analysis methods (failure events)
  - Component interaction accidents
  - Software and system design errors
  - Human errors
  - Entire socio-technical system (not just technical part)

Safety as a Dynamic Control Problem (STAMP)

- Hazards result from lack of enforcement of safety constraints in system design and operations
- Goal is to control the behavior of the components and systems as a whole to ensure safety constraints are enforced in the operating system
- A change in emphasis:

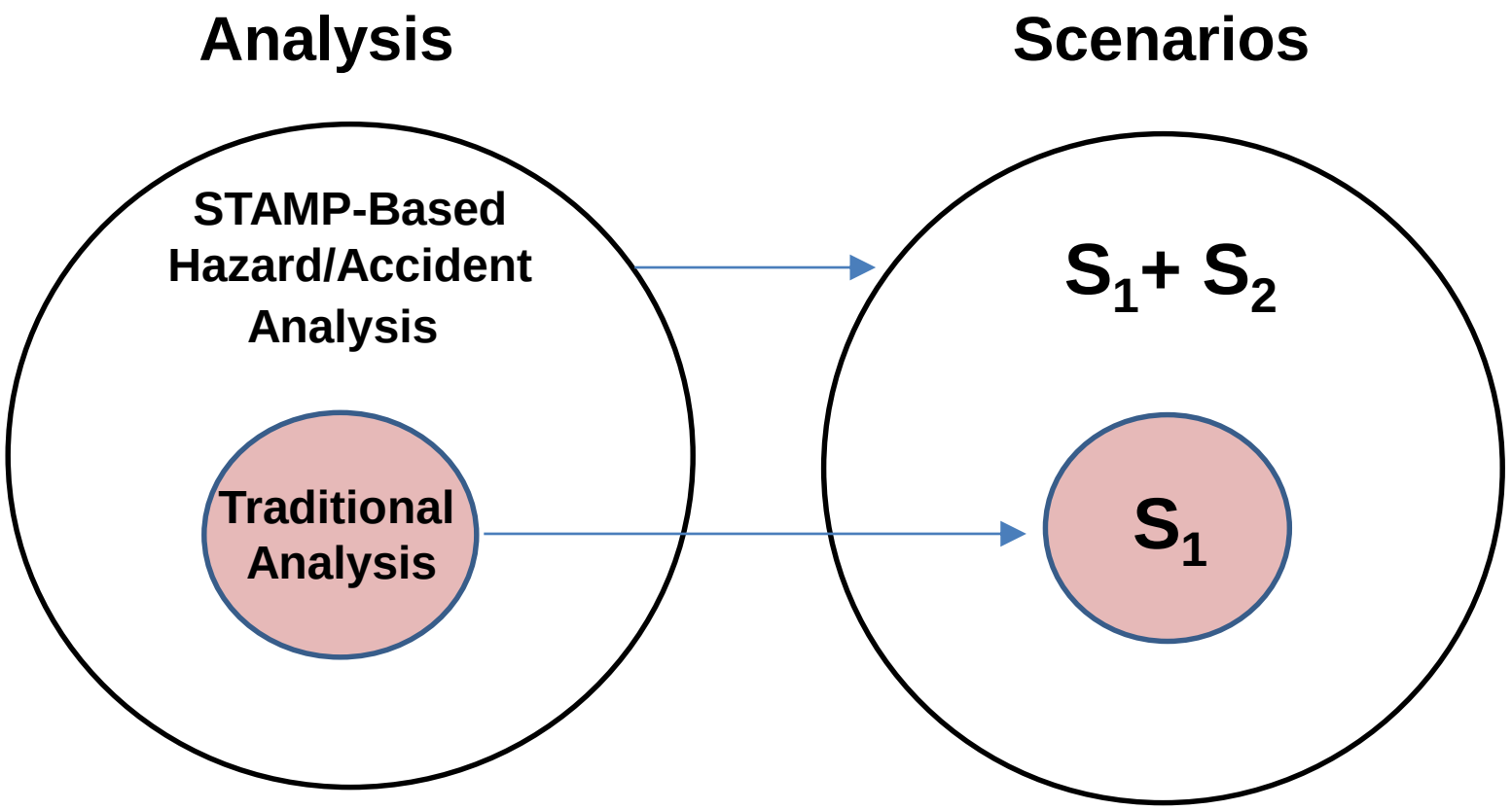


~~Increase component reliability (prevent failures)~~

Enforce safety/security constraints on system behavior

(note that enforcing constraints might require preventing failures or handling them but includes more than that)

STAMP-Based vs. Traditional Analysis

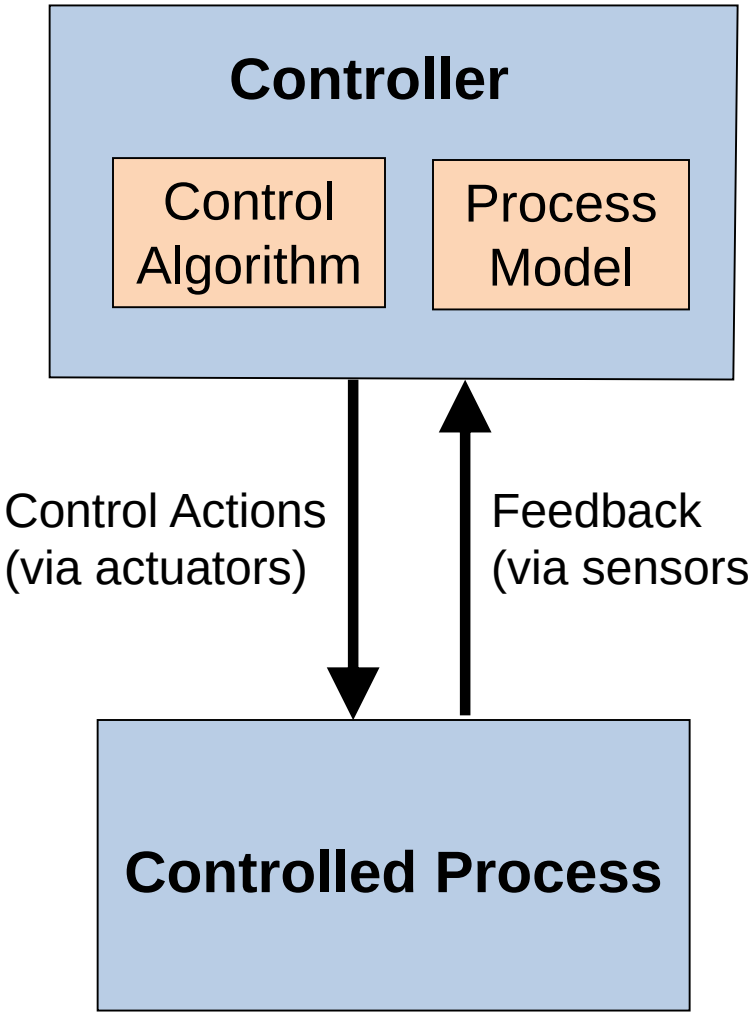


Safety as a Control Problem

Goal: Design an effective control structure that eliminates or reduces adverse events.

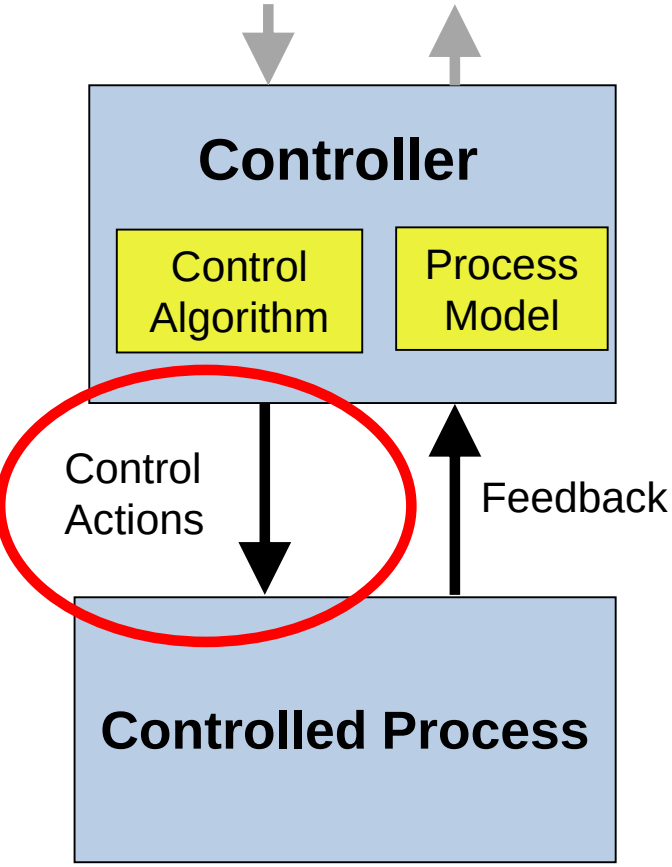
- Need clear definition of expectations, responsibilities, authority, and accountability at all levels of safety control structure
- Need appropriate feedback
- Entire control structure must together enforce the system safety property (constraints)
  - Physical design (inherent safety)
  - Operations
  - Management
  - Social interactions and culture
- Hazard/risk analysis is done on the control structure

Role of Process Models in Control



- Controllers use a **process model** to determine control actions
- Software/human related accidents often occur when the process model is incorrect
- Captures software errors, human errors, flawed requirements …

Unsafe Control Actions

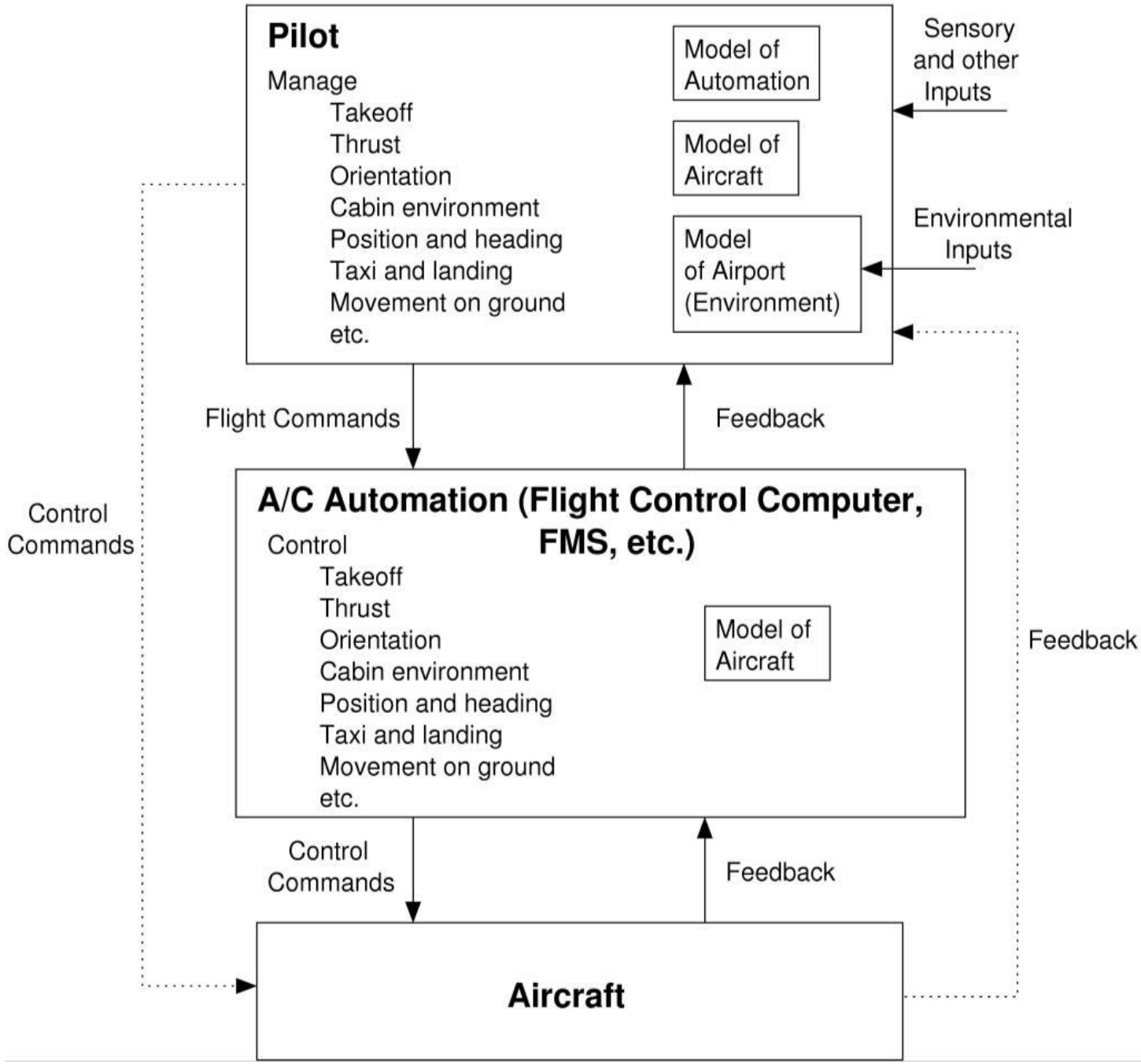


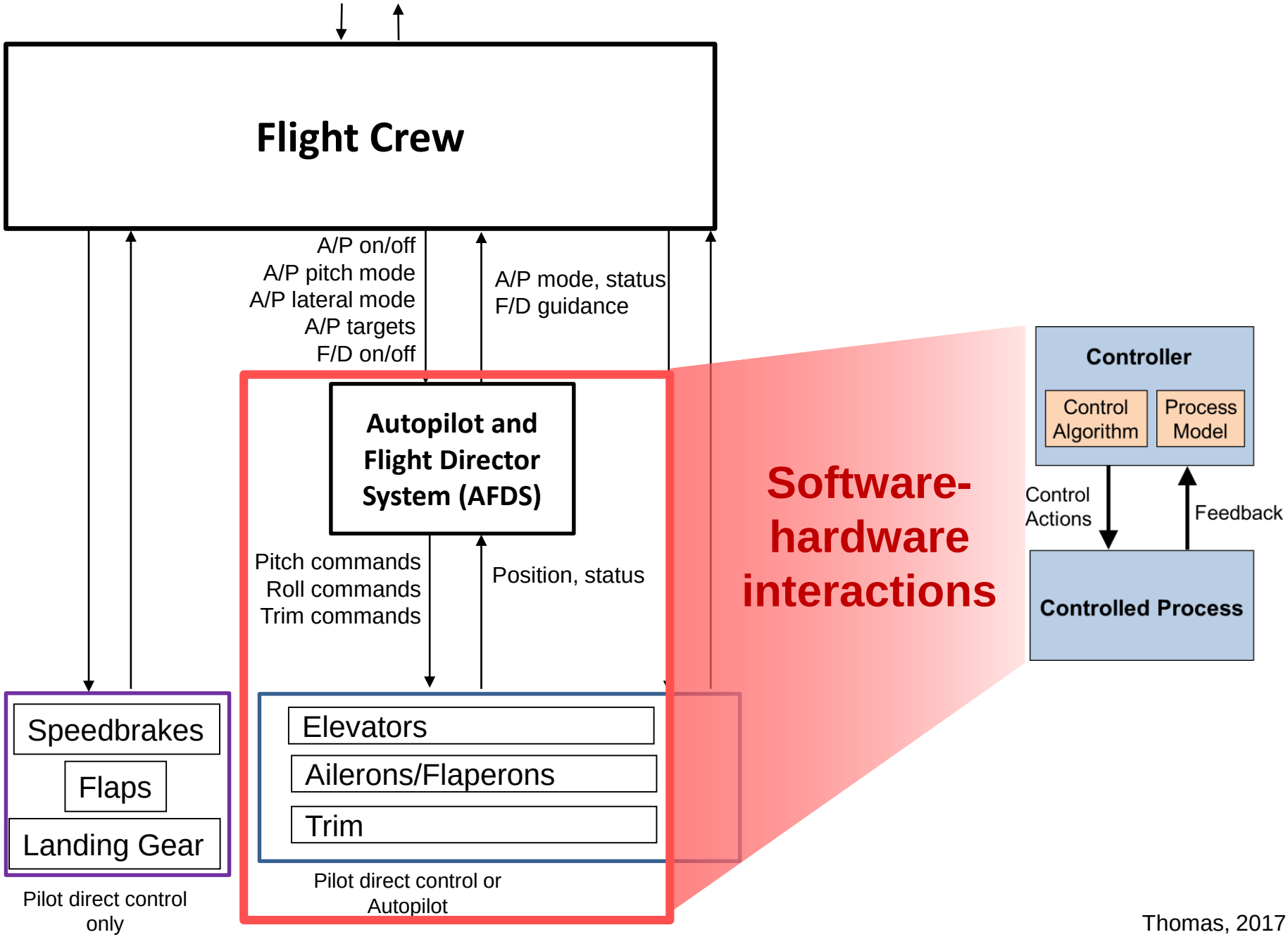
Four types of unsafe control actions

1) Control commands required for safety are not given
2) Unsafe commands are given
3) Potentially safe commands but given too early, too late
4) Control action stops too soon or applied too long (continuous control)

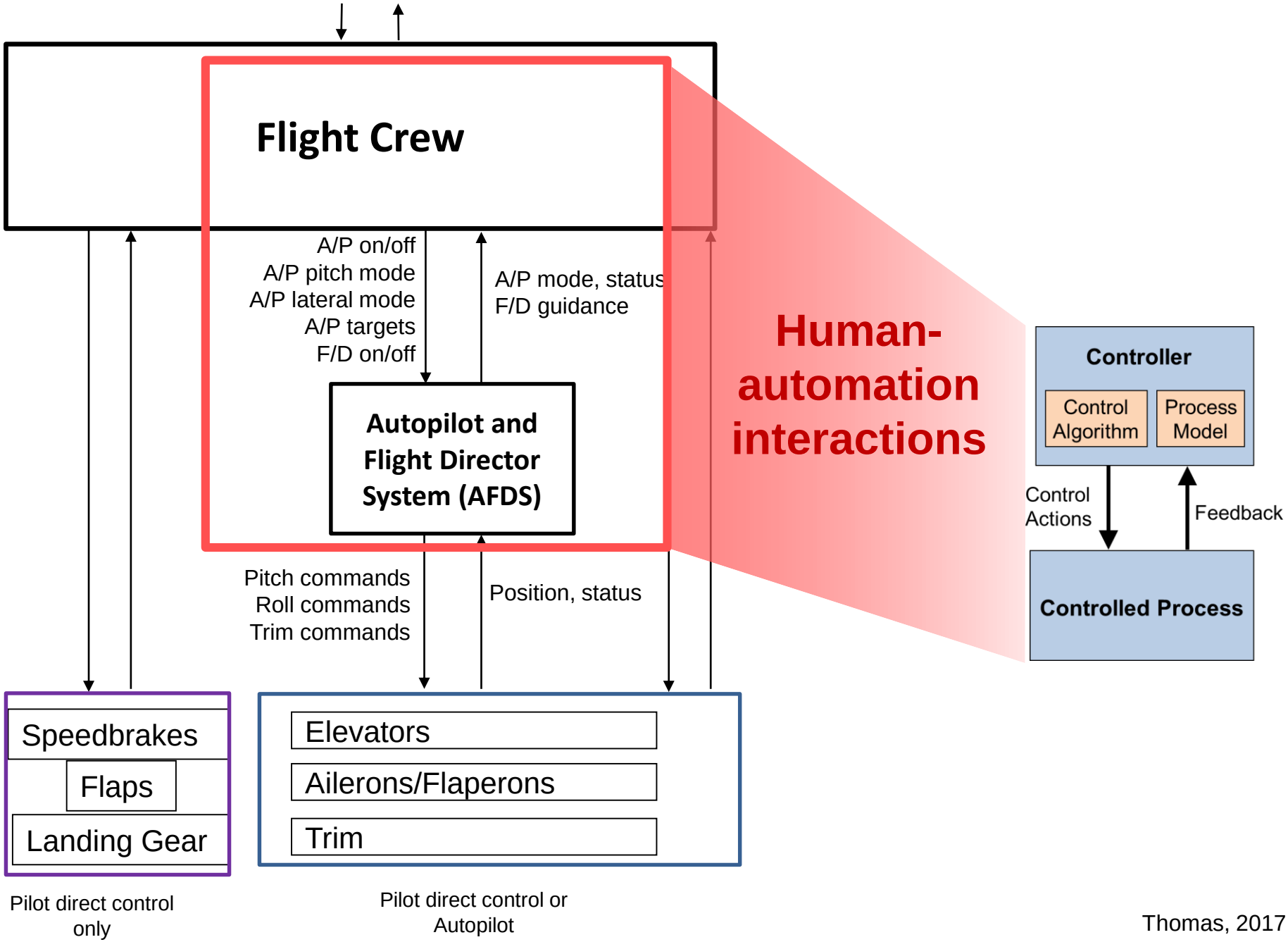
Analysis:

1. Identify potential unsafe control actions
2. Identify why they might be given
3. If safe ones provided, then why not followed?

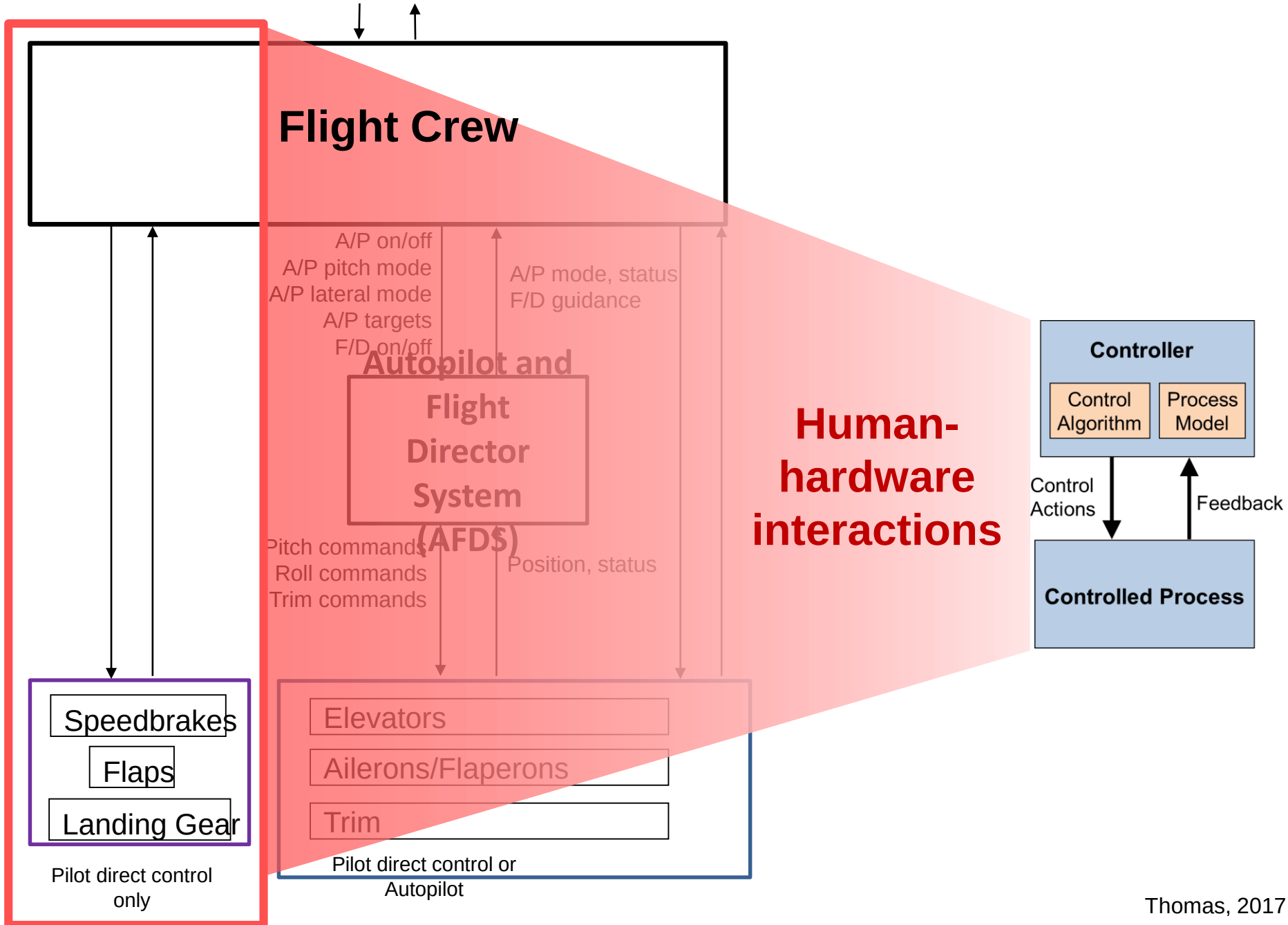

Pilot
Manage
Takeoff
Thrust
Orientation
Cabin environment
Position and heading
Taxi and landing
Movement on ground
etc.
Model of Automation
Model of Aircraft
Model of Airport (Environment)
Sensory and other Inputs
Environmental Inputs
Flight Commands
Feedback
A/C Automation (Flight Control Computer, FMS, etc.)
Control
Takeoff
Thrust
Orientation
Cabin environment
Position and heading
Taxi and landing
Movement on ground
etc.
Model of Aircraft
Control Commands
Feedback
Control Commands
Feedback
Aircraft


Flight Crew
A/P on/off
A/P pitch mode
A/P lateral mode
A/P targets
F/D on/off
A/P mode, status
F/D guidance
Autopilot and Flight Director System (AFDS)
Pitch commands
Roll commands
Trim commands
Position, status
Speedbrakes
Flaps
Landing Gear
Elevators
Ailerons/Flaperons
Trim
Pilot direct control only
Pilot direct control or Autopilot
Software-hardware interactions
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process

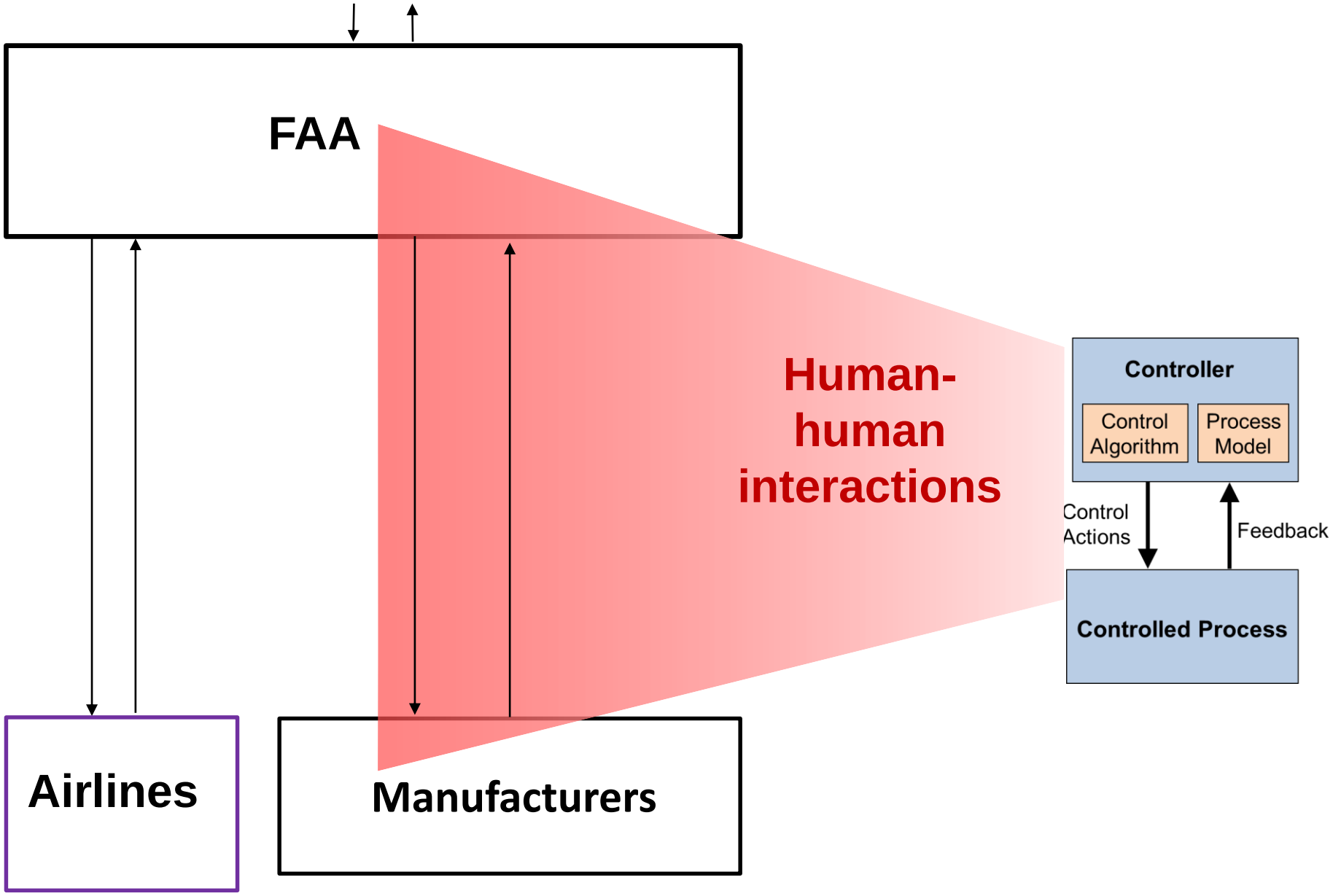
Thomas, 2017


Flight Crew
A/P on/off
A/P pitch mode
A/P lateral mode
A/P targets
F/D on/off
A/P mode, status
F/D guidance
Autopilot and Flight Director System (AFDS)
Pitch commands
Roll commands
Trim commands
Position, status
Speedbrakes
Flaps
Landing Gear
Elevators
Ailerons/Flaperons
Trim
Pilot direct control only
Pilot direct control or Autopilot
Human-automation interactions
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process

Thomas, 2017


Flight Crew
A/P on/off
A/P pitch mode
A/P lateral mode
A/P targets
F/D on/off
A/P mode, status
F/D guidance
Autopilot and Flight Director System (AFDS)
Pitch commands
Roll commands
Trim commands
Position, status
Human-hardware interactions
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process
Speedbrakes
Flaps
Landing Gear
Pilot direct control only
Elevators
Ailerons/Flaperons
Trim
Pilot direct control or Autopilot

Thomas, 2017


FAA
Human-
human
interactions
Controller
Control
Algorithm
Process
Model
Control
Actions
Feedback
Controlled Process
Airlines
Manufacturers

Thomas, 2017

STAMP:
(System-Theoretic Accident Model and Processes)

STAMP

- A new, more powerful accident/loss causality model
- Based on systems theory, not reliability theory
- Treats accidents/losses as a dynamic control problem (vs. a failure problem)
- Applies to very complex systems
- Includes
  - Scenarios from traditional hazard analysis methods (failure events)
  - Component interaction accidents
  - Software and system design errors
  - Human errors
  - Entire socio-technical system (not just technical part)

Applying Systems Theory to Safety

- Accidents involve a complex, dynamic "process"
  - Not simply chains of failure events
  - Arise in interactions among humans, machines and the environment
- Treat safety as a dynamic control problem
  - Safety requires enforcing a set of constraints on system behavior
  - Accidents occur when individual component behavior and interactions among system components violate those constraints
  - Safety becomes a control problem rather than just a reliability problem

Safety as a Dynamic Control Problem

- Examples
  - O-ring did not control propellant gas release by sealing gap in field joint of Challenger Space Shuttle
  - Software did not adequately control descent speed of Mars Polar Lander
  - In B-787 Lithium-ion batteries, system did not control interactions among components necessary to keep smoke out of aircraft
  - In DWH, did not control the pressure in the well;
  - In Navy missile system, did not control the inadvertent release of a non-dummy missile during a test
  - In 2008 financial crisis, financial system did not adequately control the use of financial instruments

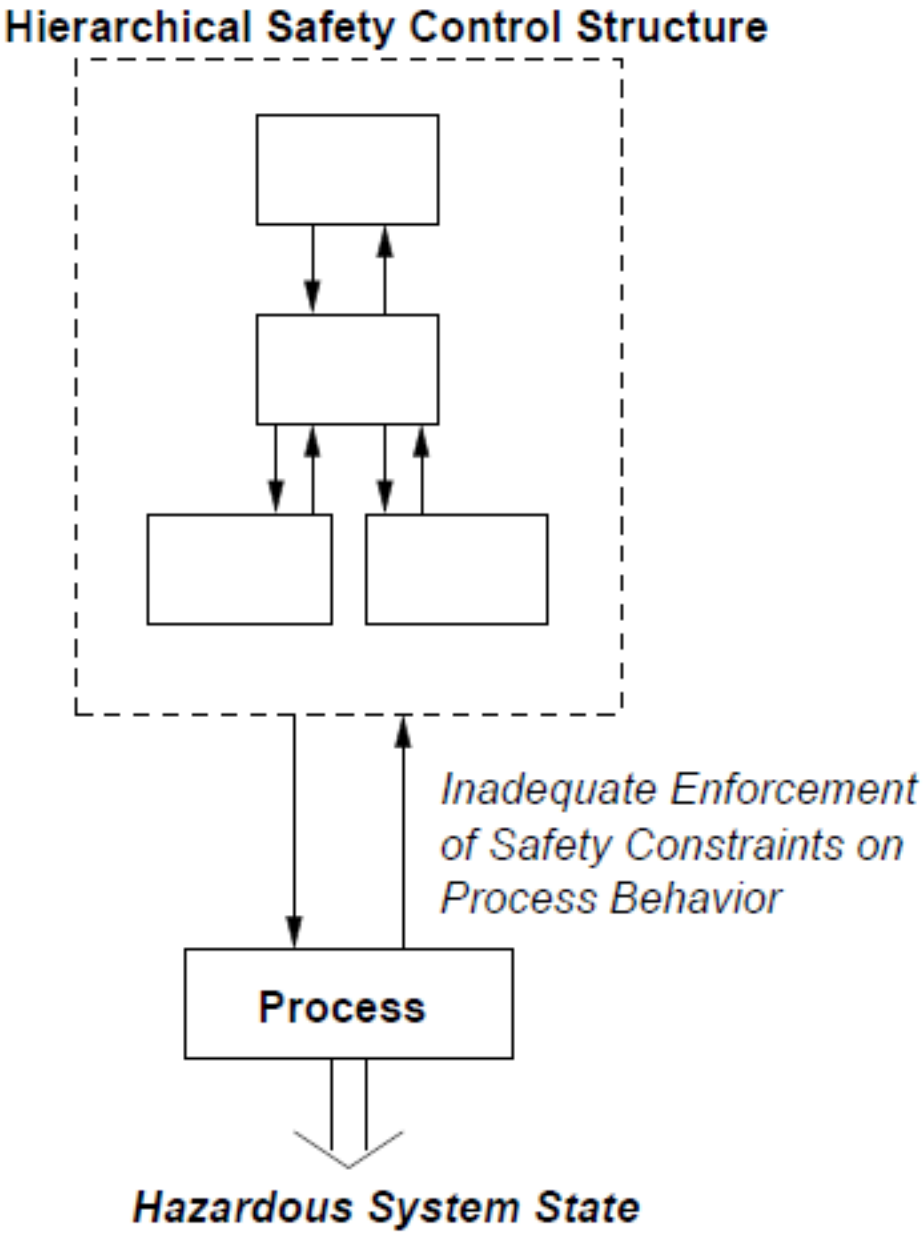
Safety as a Dynamic Control Problem (STAMP)

- Hazards and events result from lack of enforcement of safety constraints in system design and operations
- Goal is to control the behavior of the components and systems as a whole to ensure safety constraints are enforced in the operating system
- A change in emphasis:

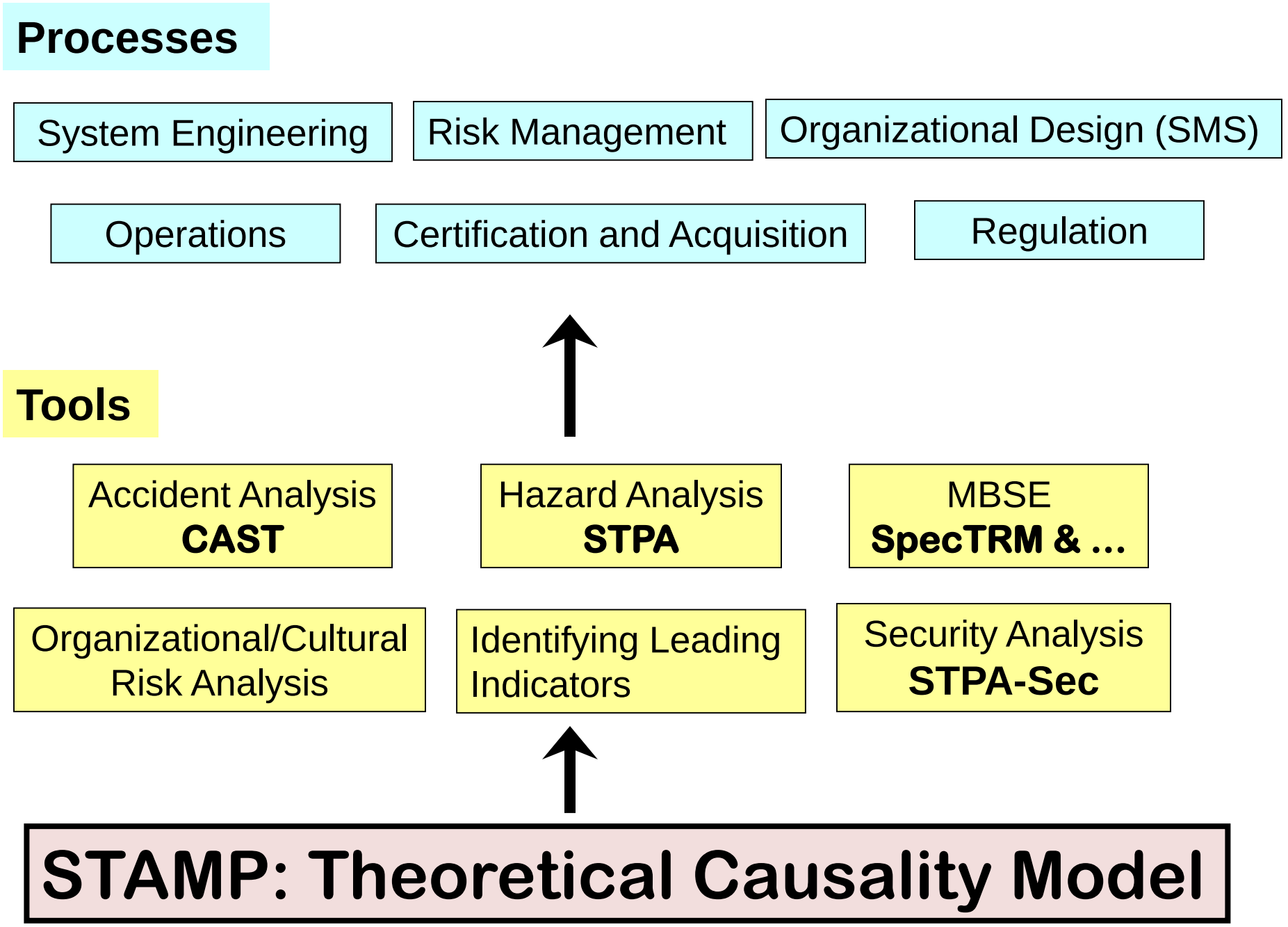
“enforce safety/security constraints on system behavior”

(note that enforcing constraints might require preventing failures or handling them but includes more than that)

Accident Causality Using STAMP



What kinds of tools are available?


Processes
System Engineering
Risk Management
Organizational Design (SMS)
Operations
Certification and Acquisition
Regulation
Tools
Accident Analysis
CAST
Hazard Analysis
STPA
MBSE
SpecTRM & …
Organizational/Cultural
Risk Analysis
Identifying Leading
Indicators
Security Analysis
STPA-Sec
STAMP: Theoretical Causality Model

BACKUP

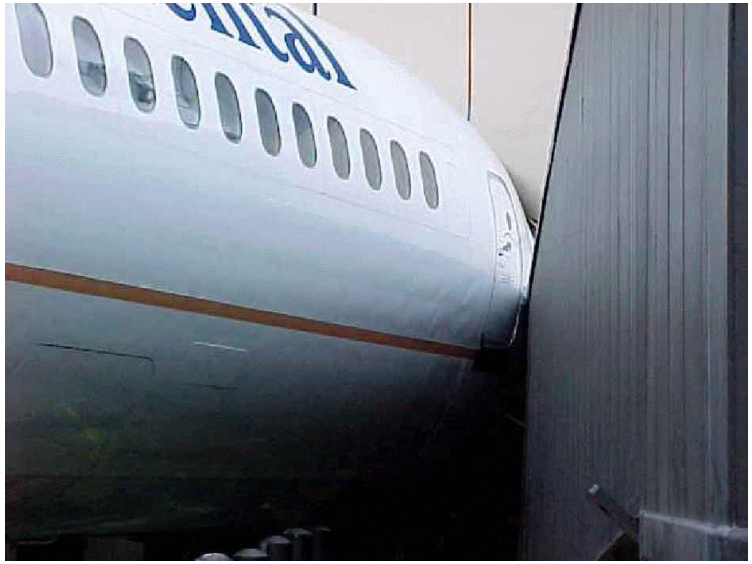
Human Factors and Aircraft Risk Assessment Today (SAE ARP 4761)

- FHA (Fault Tree analysis or FMEA or …)
  - Hardware and functions only
  - Based on probabilistic analysis
- Software handled separately
  - Software assurance standard (testing)
  - Ignores software requirements
- Human factors handled separately
  - Focus mostly on cockpit design

From SAE ARP 4761

Function	Failure Condition (Hazard Description)	Phase	Effect of Failure Condition on Aircraft/Crew	Classification
Decelerate Aircraft on the Ground	Loss of Deceleration Capability	Landing/ RTO/ Taxi		
	…	…	…	
	c. Unannunciated loss of deceleration capability	Taxi	Crew is unable to stop the aircraft on the taxi way or gate resulting In low speed contact with terminal, aircraft, or vehicles	Major
	d. Annunciated loss of deceleration capability	Taxi	Crew steers the aircraft clear of any obstacles and calls for a tug or portable stairs	No Safety Effect

Continental Airlines Introduces the Improved Disembarkation Method



Bottom Line

- Complexity is reaching a new level (tipping point)
  - Old approaches becoming less effective
  - New causes of mishaps appearing (especially related to use of software and autonomy)
- Traditional approaches do not provide the information necessary to prevent losses in these systems
- Need a paradigm change

Change focus

~~Increase component reliability (analytic decomposition)~~

Enforce safe behavior (dynamic control using systems theory)

Bottom Line (2)

- Allows creation of new analysis and engineering approaches
  - More powerful and inclusive
  - Orders of magnitude less expensive
  - Work on very complex systems (top-down system engineering)
  - Design safety and security and other properties in from the beginning
  - Compliant with MIL-STD-882E and other military standards

- New paradigm works better than old techniques:
  - Empirical evaluations and controlled studies show it finds more causal scenarios (the “unknown unknowns”)
  - Can be used before a detailed design exists to create safety and security requirements

Investigating/Understanding Accidents

Learning from accidents in order to prevent them in the future

- Identifying ALL the factors involved
- Identifying the causal factors

Hazard Analysis

- “Investigating an accident before it occurs”
- Identifying potential causal scenarios and using them to improve design and operations
- Worst case analysis vs. average (expected) case analysis

Design for Safety

- Eliminate or control scenarios (causal factors) identified by hazard analysis
- Design to prevent operator error
  - Human errors will occur
  - Need to make sure they don't result in an accident
  - Design so that your technology doesn't induce human error

Operations

- Operating systems to prevent and reduce accidents
- The Safety Information System
- Creating an Operational Safety Management Plan
- Identifying leading indicators for increasing risk

Management

- Why management should care about safety. Does safety conflict with productivity and profits?
- Safety culture and how a good one is created
- Designing and maintaining a safety management system

(image from http://www.thecais ow-Great-Managers-Lead)

Risk Assessment/Assurance?

- Is probabilistic risk assessment feasible?
- Even if were possible, too late to do anything about it
  - Designing to be safe vs. assuring after the fact
  - Systems too complex today to realistically change after the design is finished.