CREDC

CYBER RESILIENT ENERGY DELIVERY CONSORTIUM

Seminar Series



Engineering a Safer and More Secure World

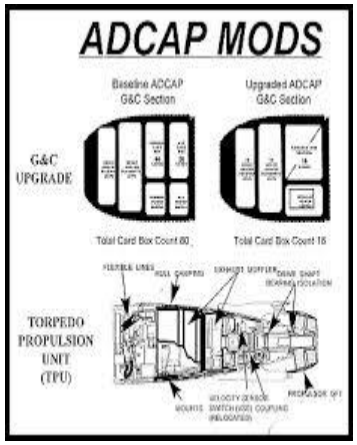
Nancy Leveson

MIT



- You’ve carefully thought out all the angles
- You’ve done it a thousand times
- It comes naturally to you
- You know what you’re doing, it’s what you’ve been trained to do your whole life.
- Nothing could possibly go wrong, right?

Think Again


ADCAP MODS
Baseline ADCAP G&C Section
Upgraded ADCAP G&C Section
G&C UPGRADE
Total Card Box Count 60
Total Card Box Count 16
TORPEDO PROPULSION UNIT (TPU)

Presentation Outline

- Complexity is reaching a new level (tipping point)
  - Old approaches becoming less effective
  - New causes of accidents appearing (especially related to use of software)
- Need a paradigm change

Change focus

~~Increase component reliability~~

Enforce safe behavior (dynamic control)

Presentation Outline (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 in from the beginning
- New paradigm applies to security too and, in fact, any emergent system property
- Does it work? Evaluations and experience so far show it works much better than what we are doing today.

Understanding The Problem

"It's never what we don't know that stops us. It's what we do know that just ain't so."

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

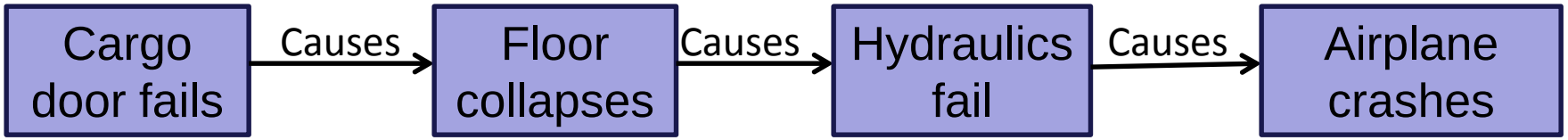
Bow Tie (CCA) FTA + ETA

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

Assumes accidents caused by component failures

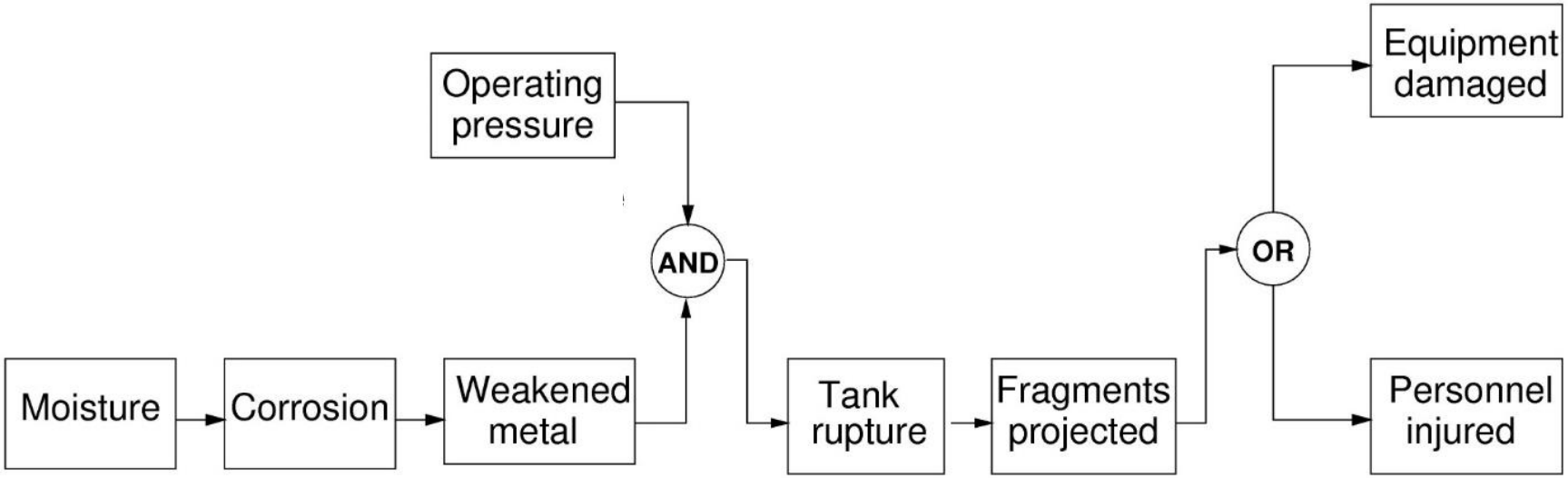
Domino “Chain of events” Model

DC-10:

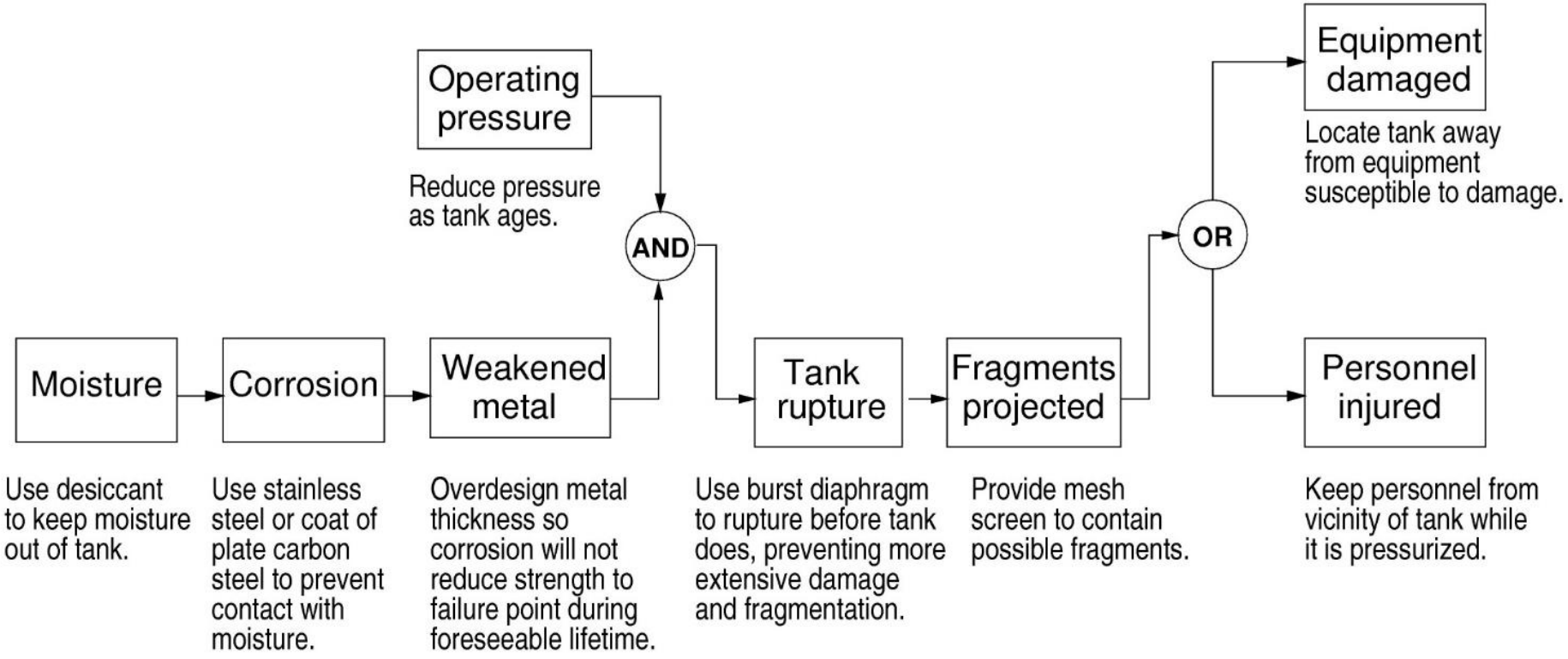


Failure Event-Based

Chain-of-events Example



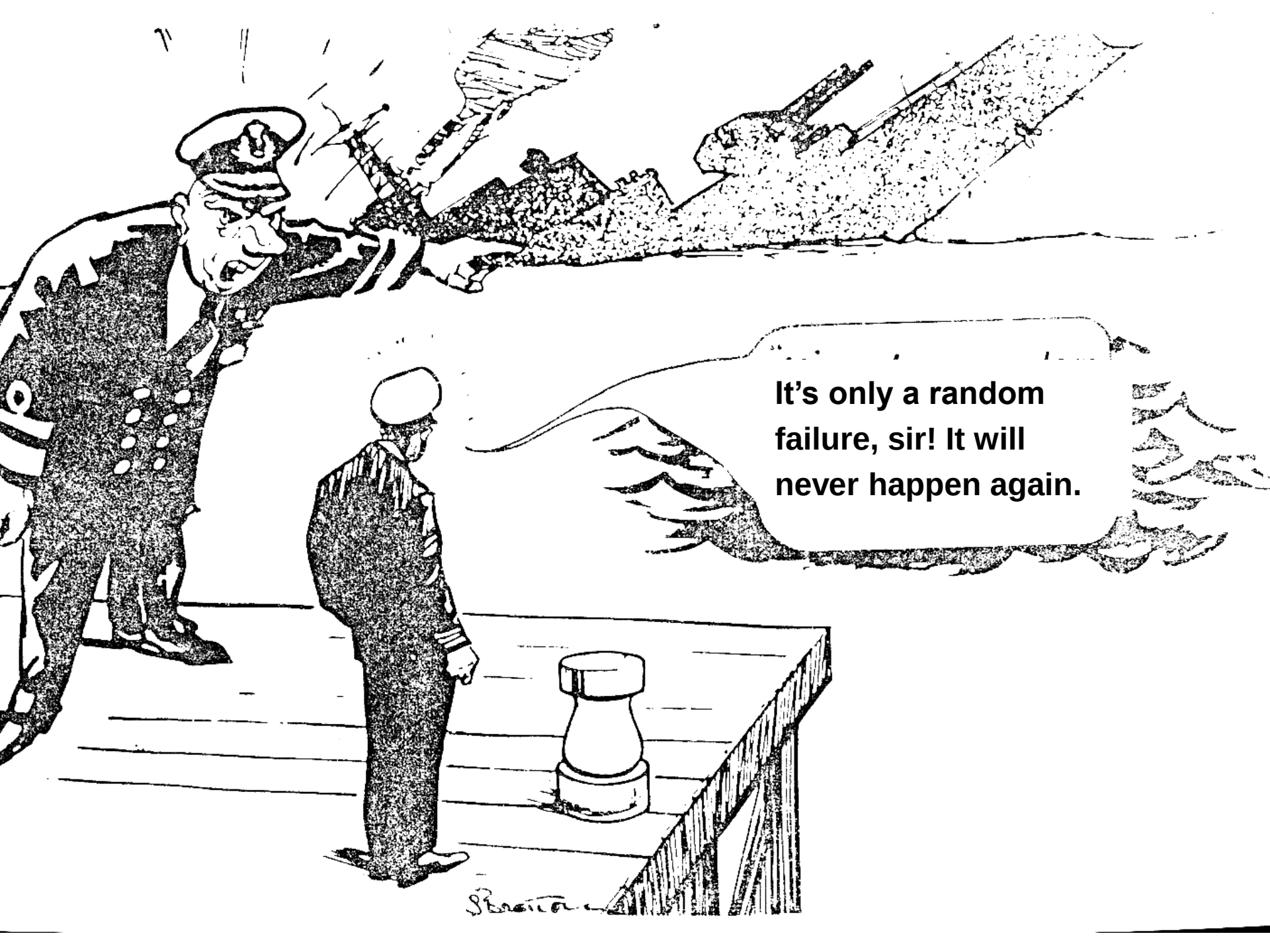
Chain-of-events Example



Traditional Approach to Safety

- Traditionally view safety as a failure problem
  - Chain of directly related failure events leads to loss
  - Establish barriers between events or try to prevent component failures
- Limitations
  - Systems are becoming more complex
    - Accidents often result from interactions among components
    - Too complex to anticipate all potential interactions
  - Omits or oversimplifies important factors
    - Human error
    - New technology
    - Culture and management
    - Evolution and adaptation

Accidents are not just the result of random failure


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

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. Cut off engines when determine have landed.
  - But “noise” (false signals) by sensors generated when parachute opens. 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

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.

Boeing 787 Lithium Battery Fires

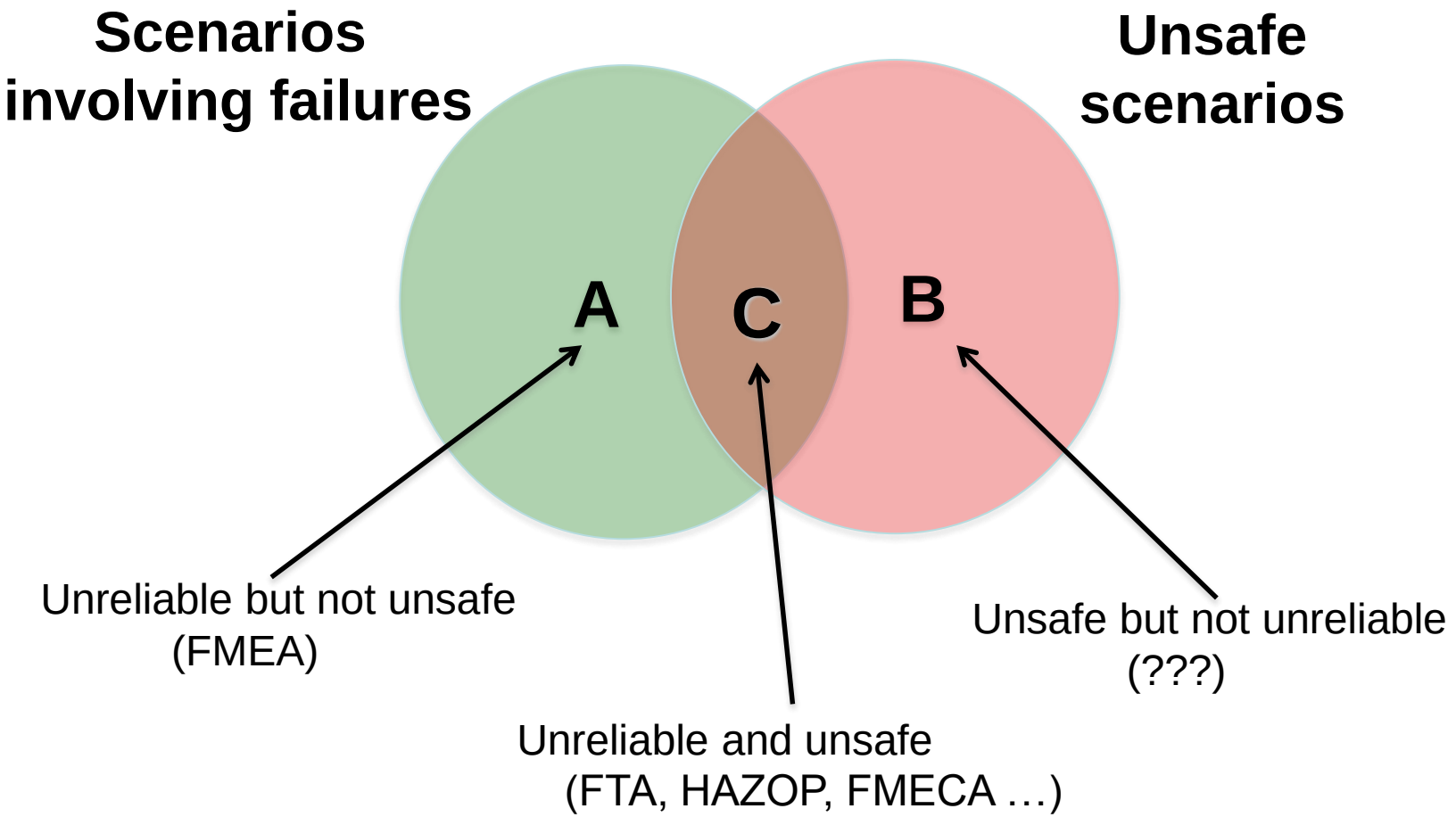
- A module monitors for smoke in the battery bay, controls fans and ducts to exhaust smoke overboard.
- Power unit monitors for low battery voltage, shut down various electronics, including ventilation
- Smoke could not be redirected outside cabin

All software requirements were satisfied!
The requirements were unsafe

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 in our system designs, which leads to system design and system engineering errors
  - No components may have “failed”

Confusing Safety and Reliability

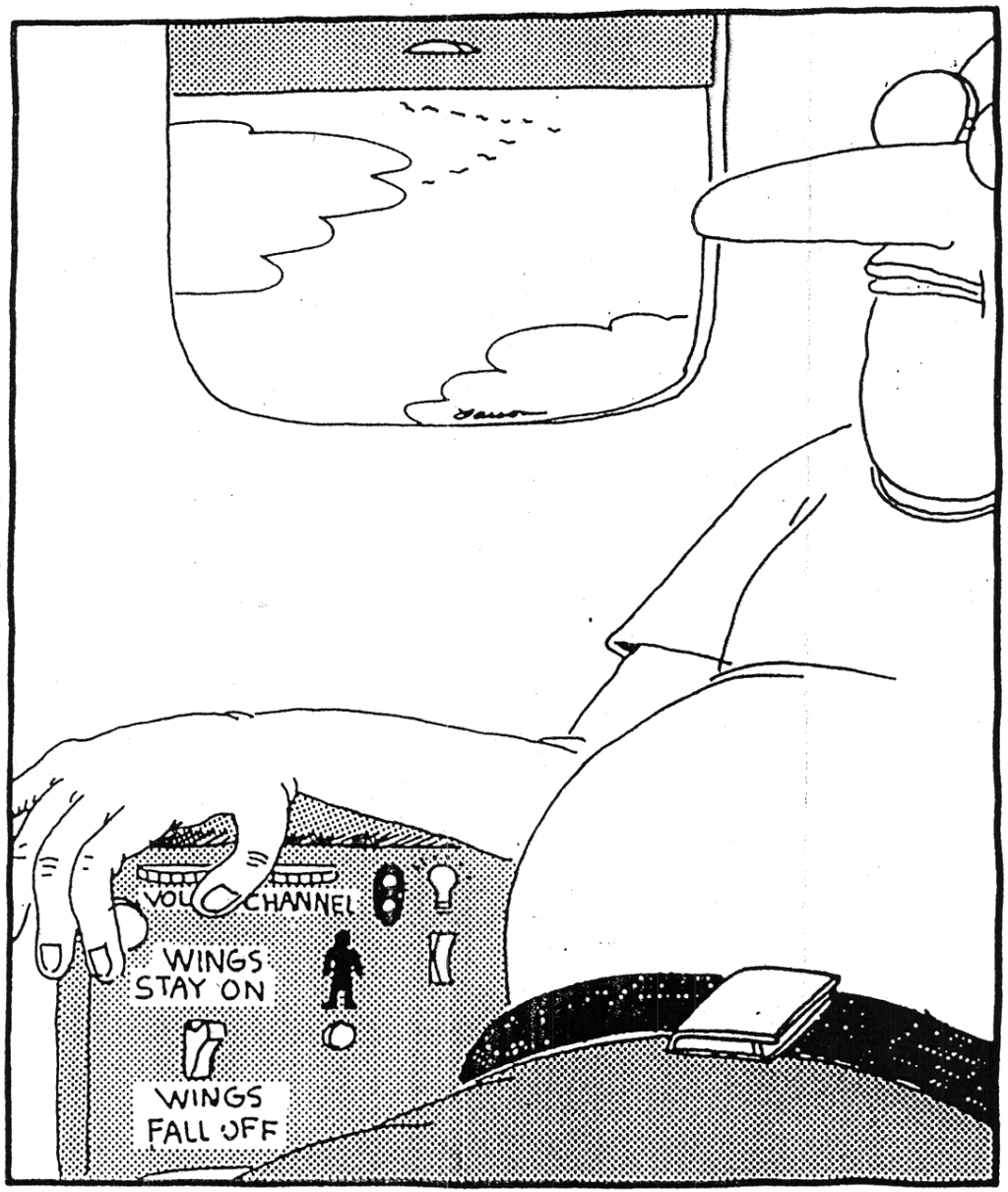


Preventing Component or Functional Failures is Not Enough

Do Operators Really Cause Most Accidents?

Operator Error: Traditional View

- Operator error is cause of most incidents and accidents
- So do something about operator involved (suspend, retrain, fire them)
- Or do something about operators in general
  - Marginalize them by putting in more automation
  - Rigidify their work by creating more rules and procedures



Fumbling for his recline button Ted unwittingly instigates a disaster

Operator Error: Systems View

- Operator error is a symptom, not a cause
- All behavior affected by context (system) in which occurs
  - Role of operators is changing in software-intensive systems as is the errors they make
  - Designing systems in which operator error inevitable and then blame accidents on operators rather than designers
- To do something about operator error, must look at system in which people work:
  - Design of equipment
  - Usefulness of procedures
  - Existence of goal conflicts and production pressures
- **Human error is a symptom of a system that needs to be redesigned**

Human factors concentrates on the “screen out”

Engineering concentrates on the “screen in”

Not enough attention on integrated system as a whole

(e.g, mode confusion, situation awareness errors, 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

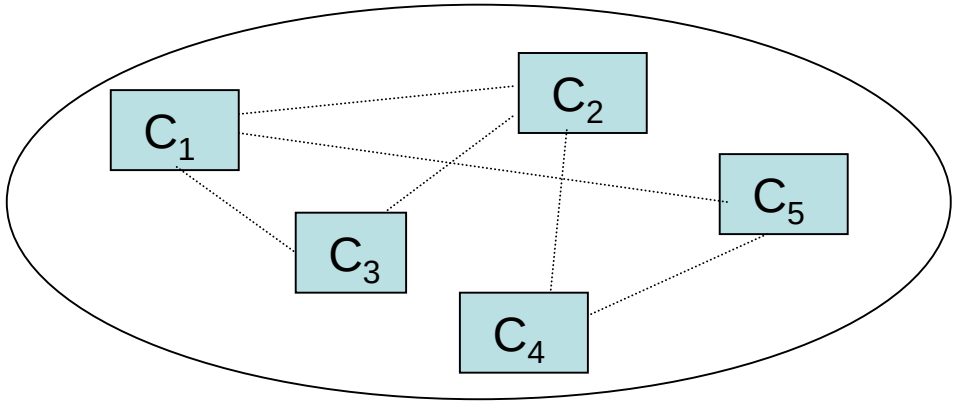
Shoehorn new technology and new levels of complexity into old methods

Traditional Approach to Coping with Complexity

Analytic Reduction ("Divide and Conquer")

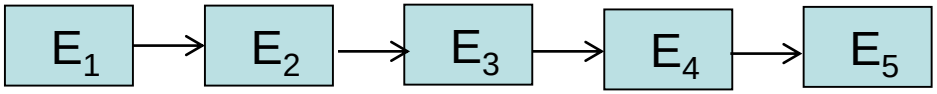
1. Divide system into separate parts

Physical/Functional: Separate into distinct components



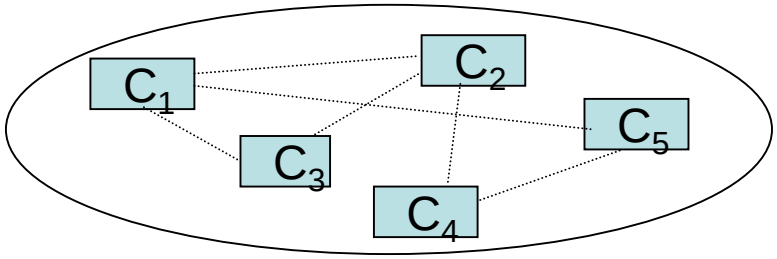
Components interact
In direct ways

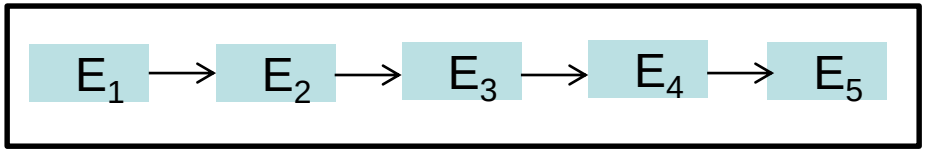
Behavior: Separate into events over time



Each event is the direct
result of the preceding event

Analytic Reduction (2)





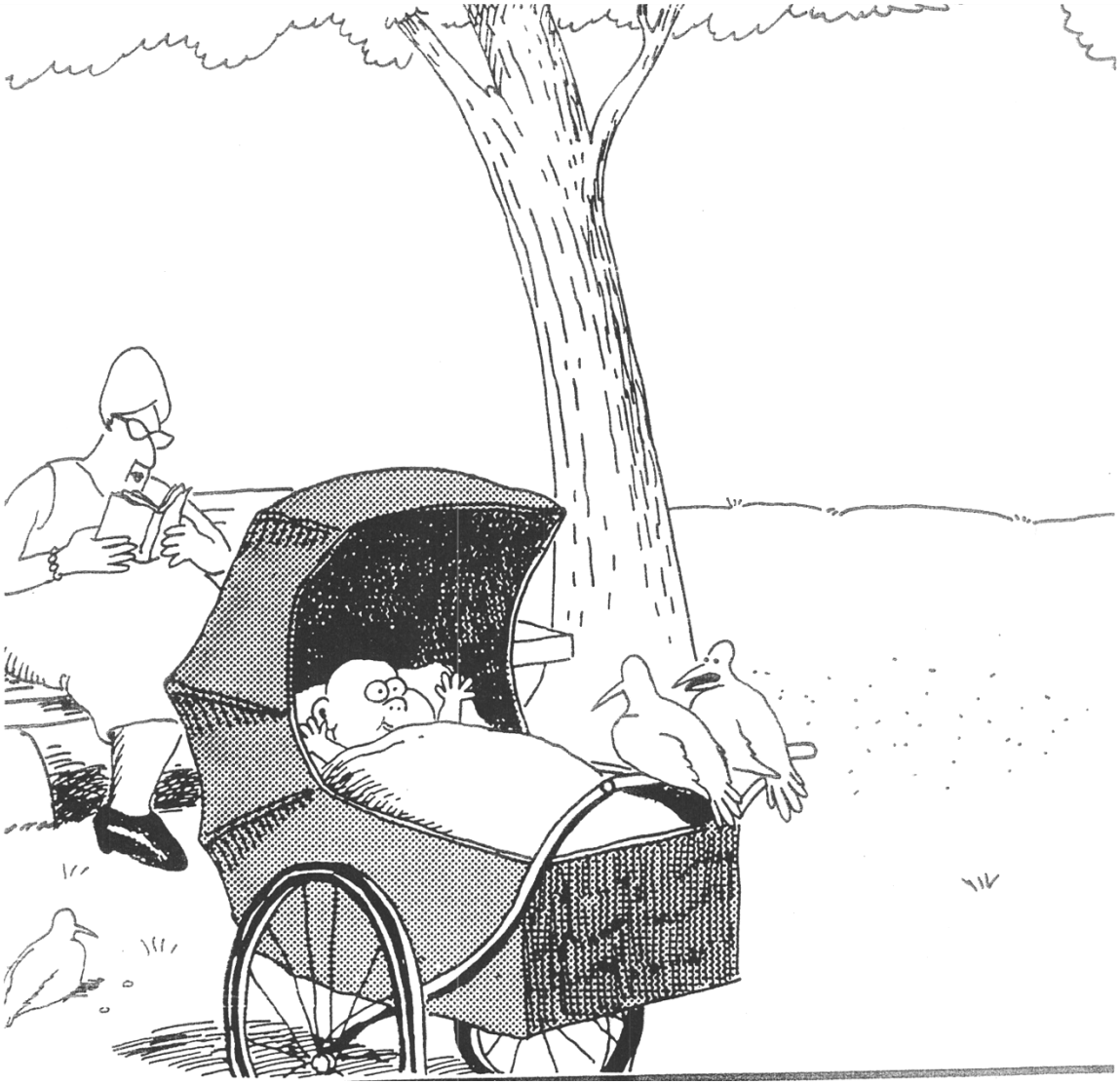
2. Analyze/examine pieces separately and combine results

- 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

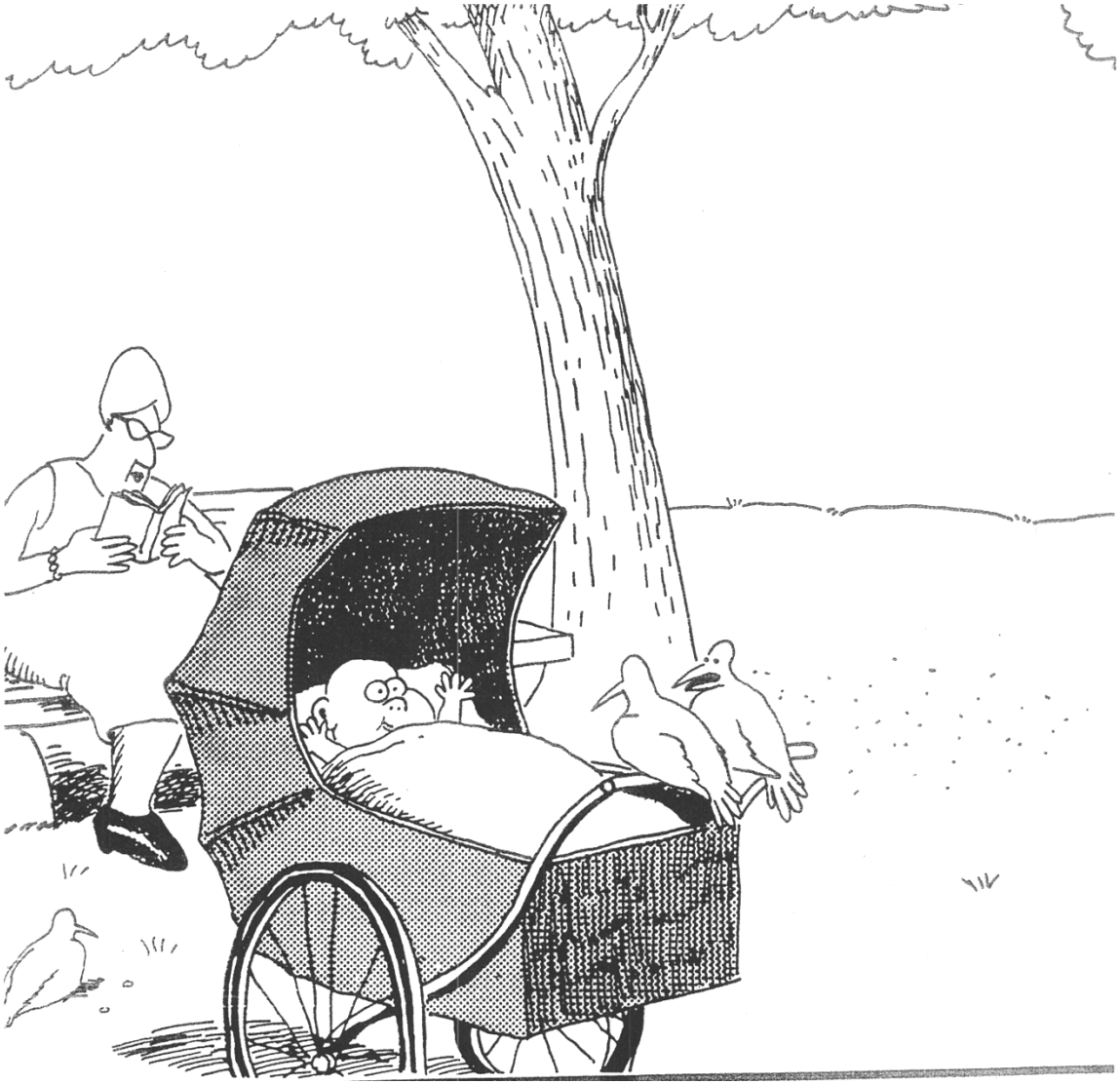
Bottom Line

- 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



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

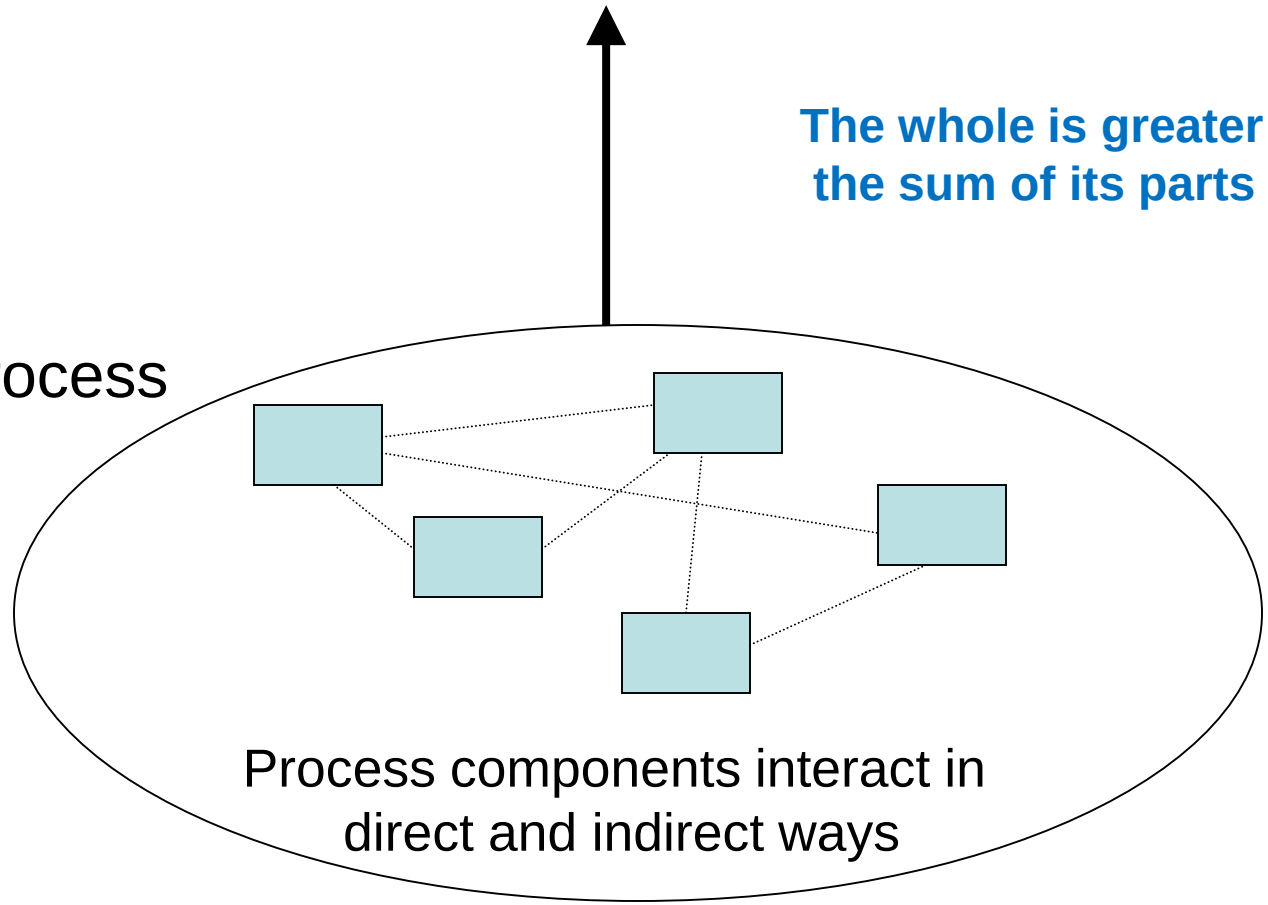


It’s still hungry … and I’ve been stuffing worms into it all day.

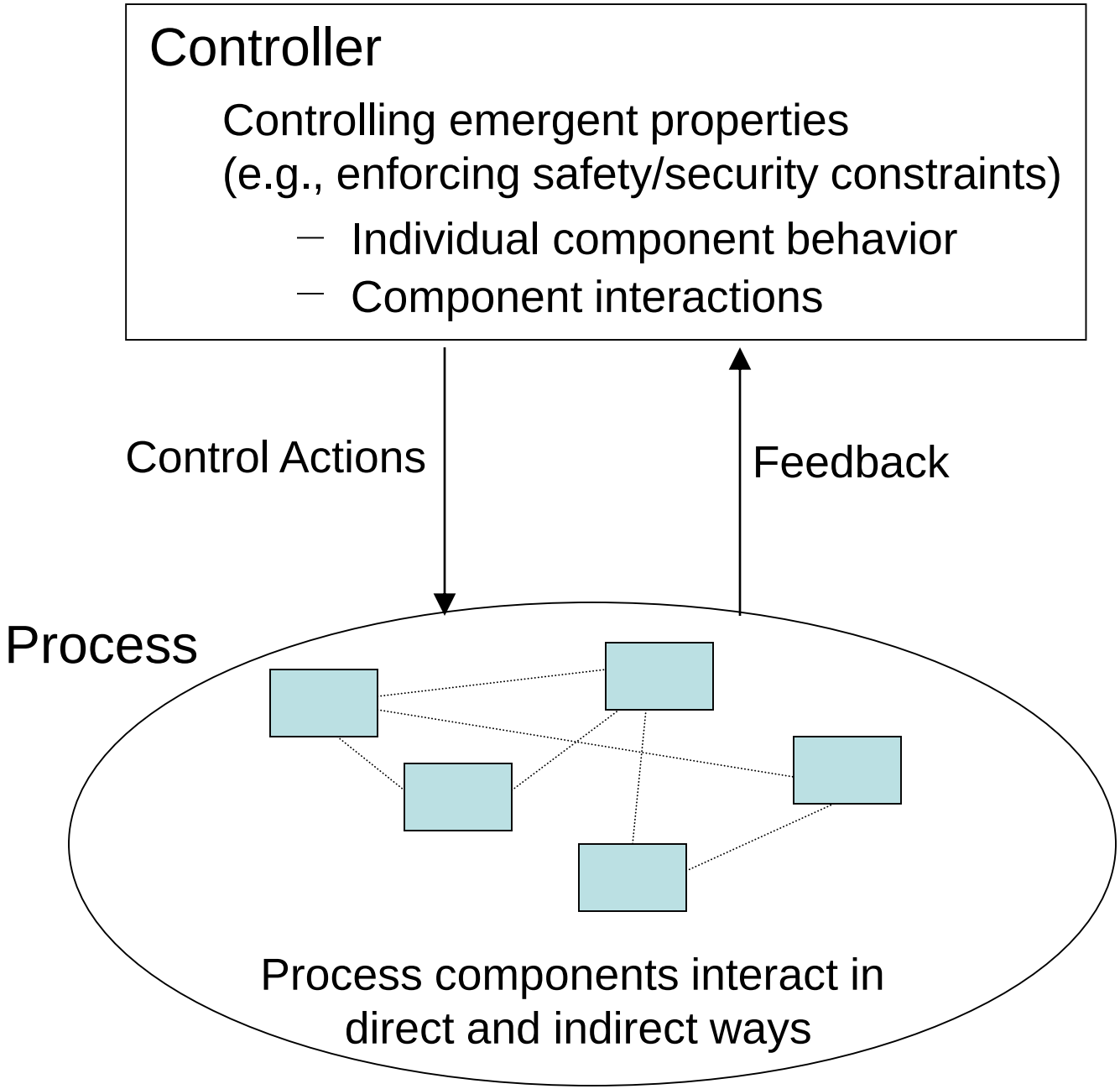
We Need New Tools for the New Problems

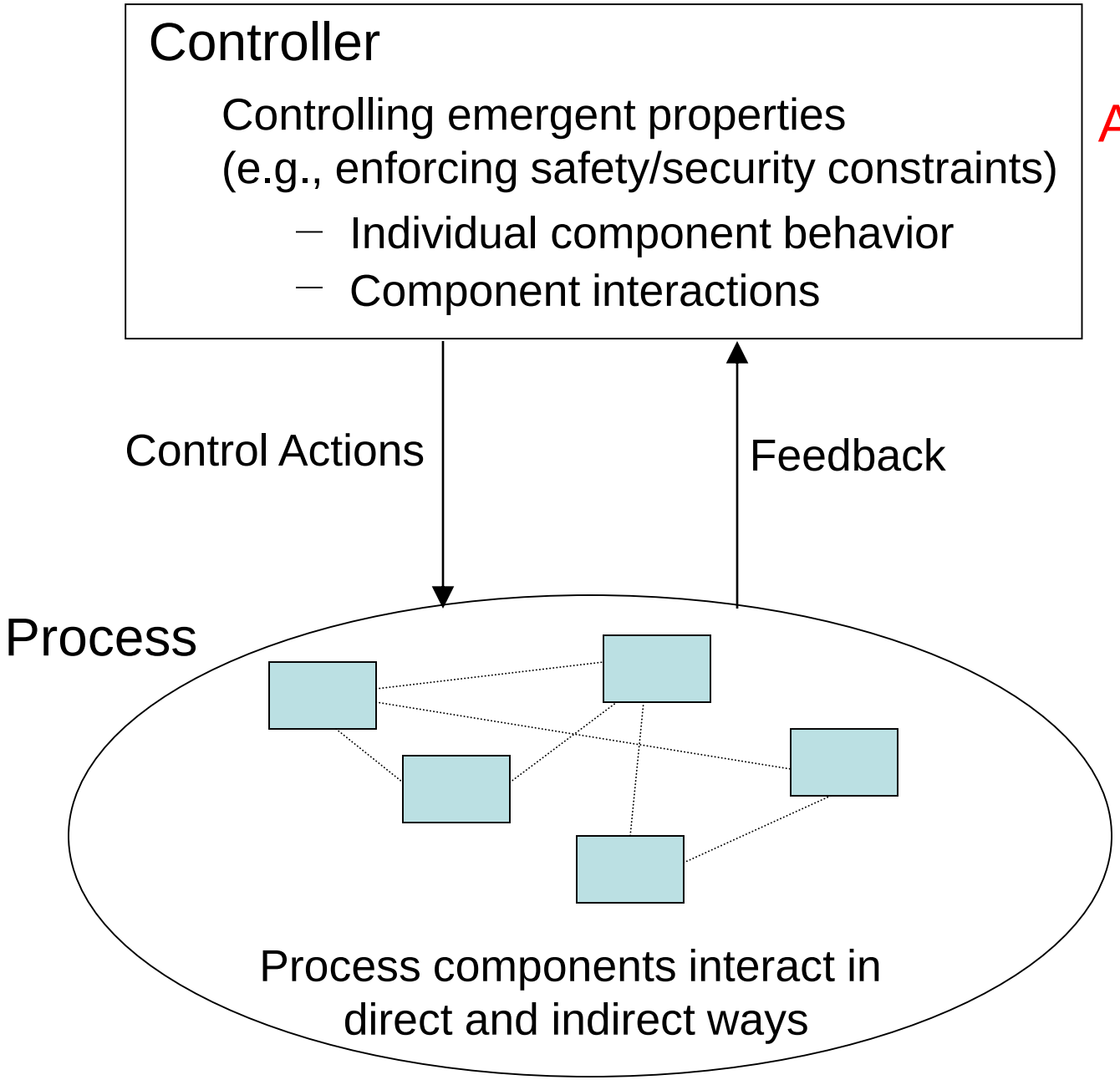
A Systems Theoretic View of Safety and Cyber Security

Emergent properties
(arise from complex interactions)



Safety and security are emergent properties


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



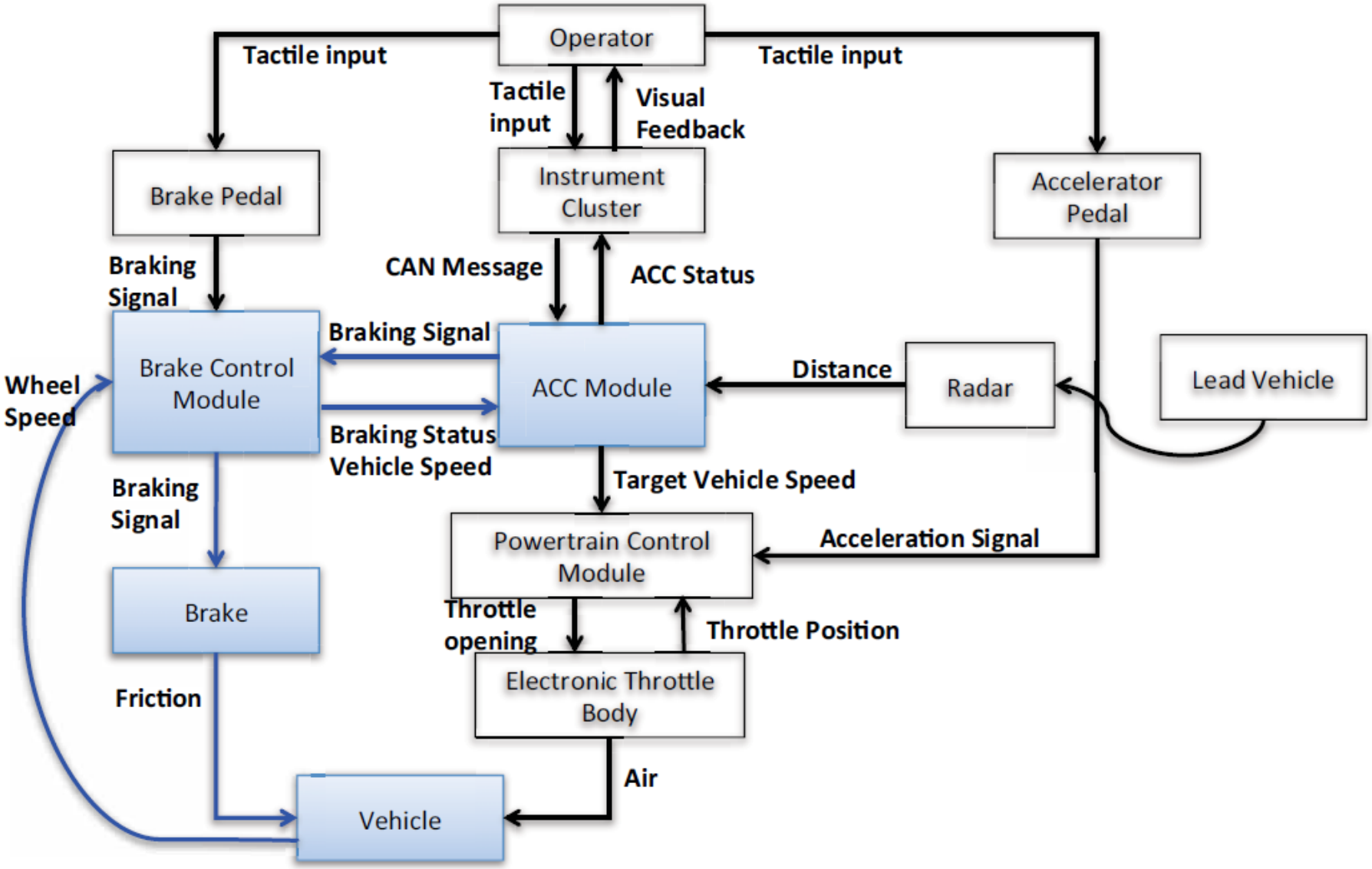
Air Traffic Control:
Safety
Throughput

Controls/Controllers Enforce Safety Constraints

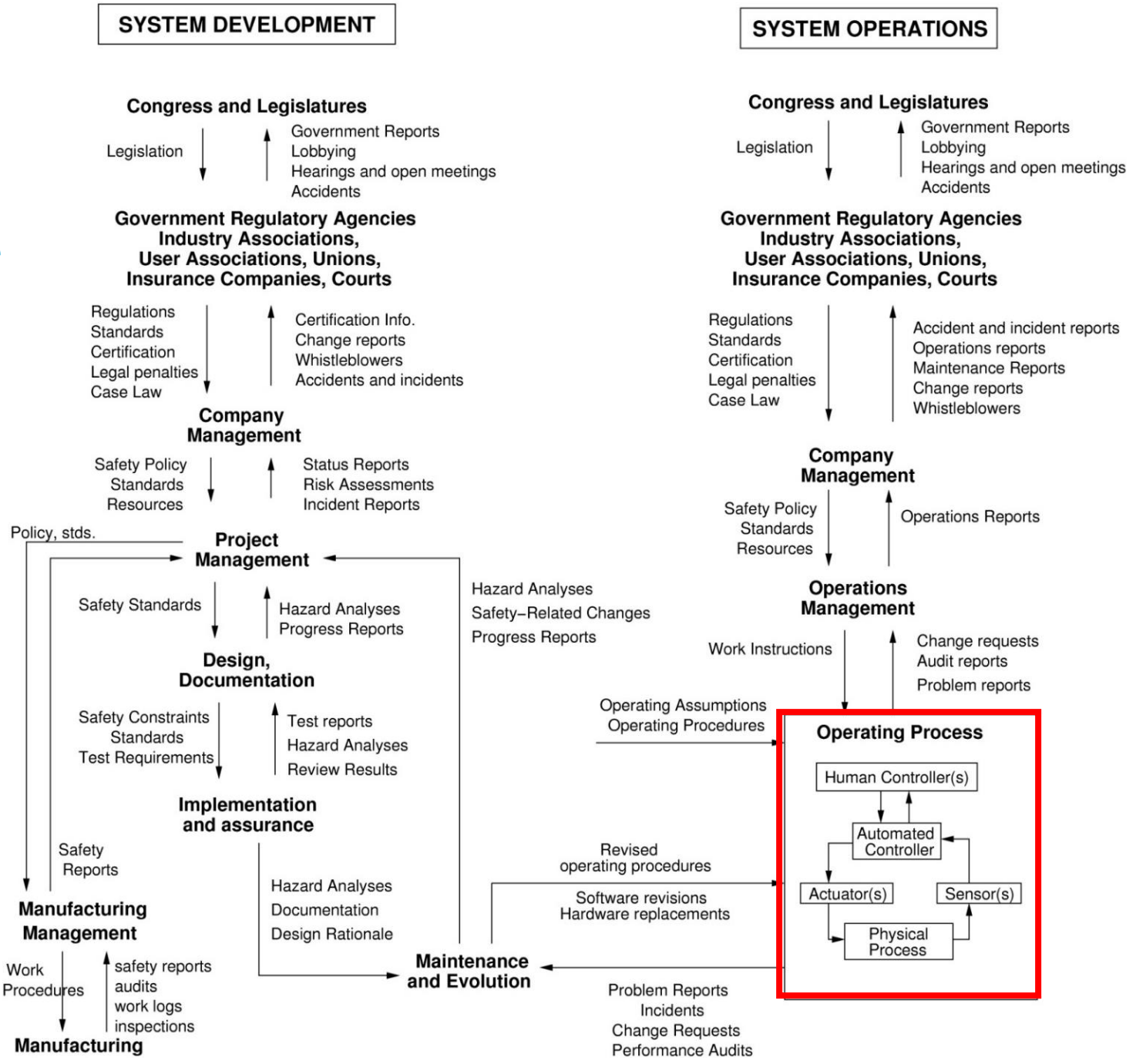
- Two aircraft/automobiles must not violate minimum separation
- Aircraft must maintain sufficient lift to remain airborne
- Level of liquid in an ISOM tower must remain below a specified level
- Toxic chemicals/radiation must not be released from plant
- Pressure in a deep water well must always be controlled
- Weapons must never be detonated inadvertently

These are the High-Level Functional Safety Requirements (What/Why) to Address During Design (How)

Example: Adaptive Cruise Control (ACC)



Example Safety Control Structure



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

A Broad View of “Control”

Component failures and unsafe interactions may be “controlled” through design

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

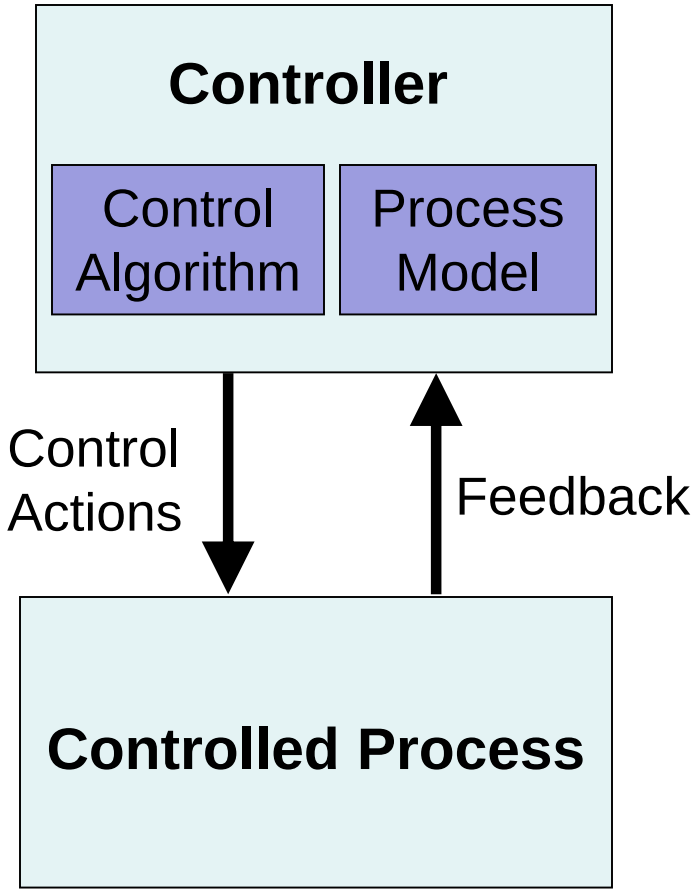
or through process

- 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)

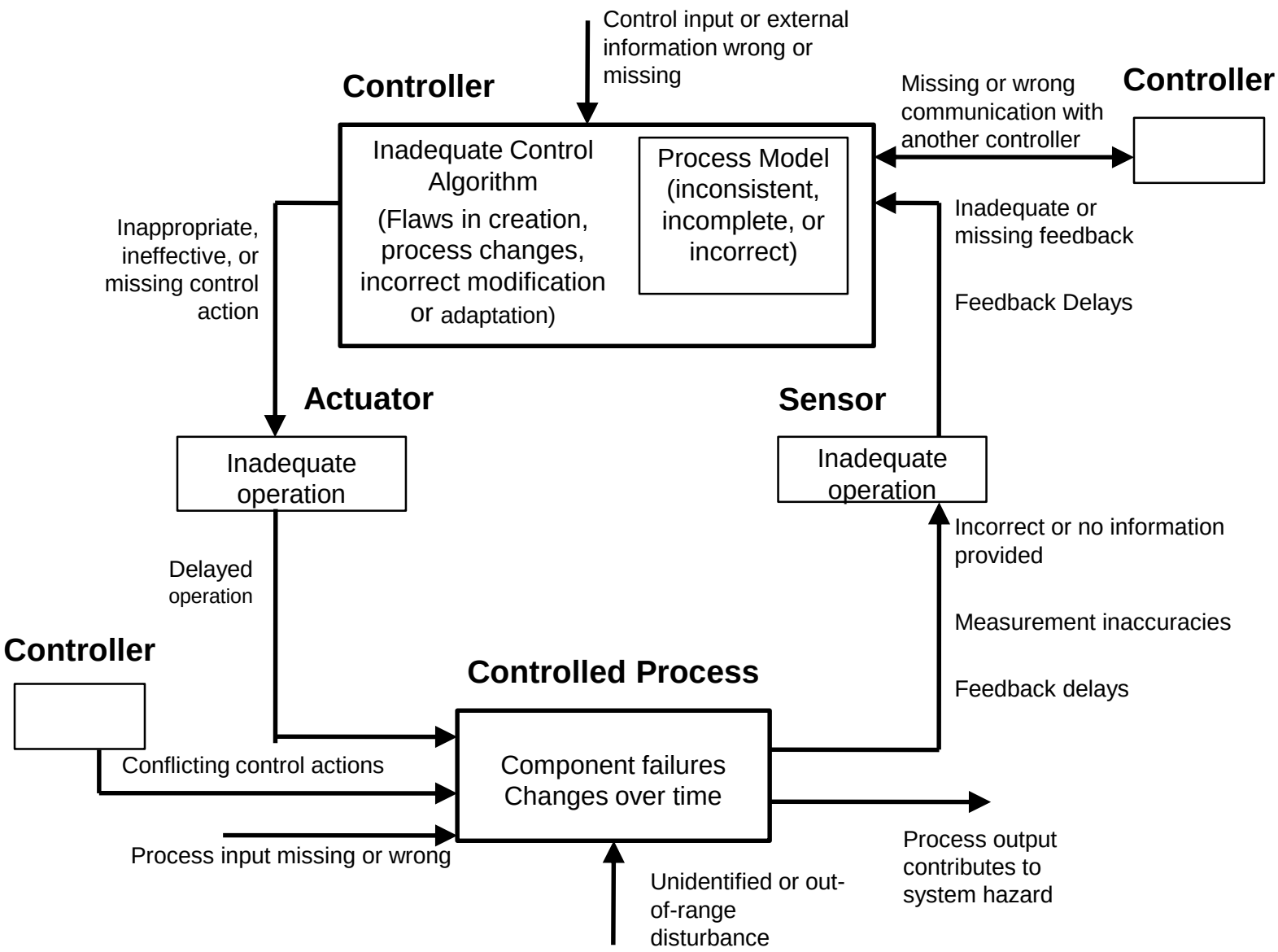
Role of Process Models in Control



- Controllers use a **process model** to determine control actions
- Accidents often occur when the process model is incorrect
- Four types of unsafe control actions:
  - Control commands required for safety are not given
  - Unsafe ones are given
  - Potentially safe commands given too early, too late
  - Control stops too soon or applied too long

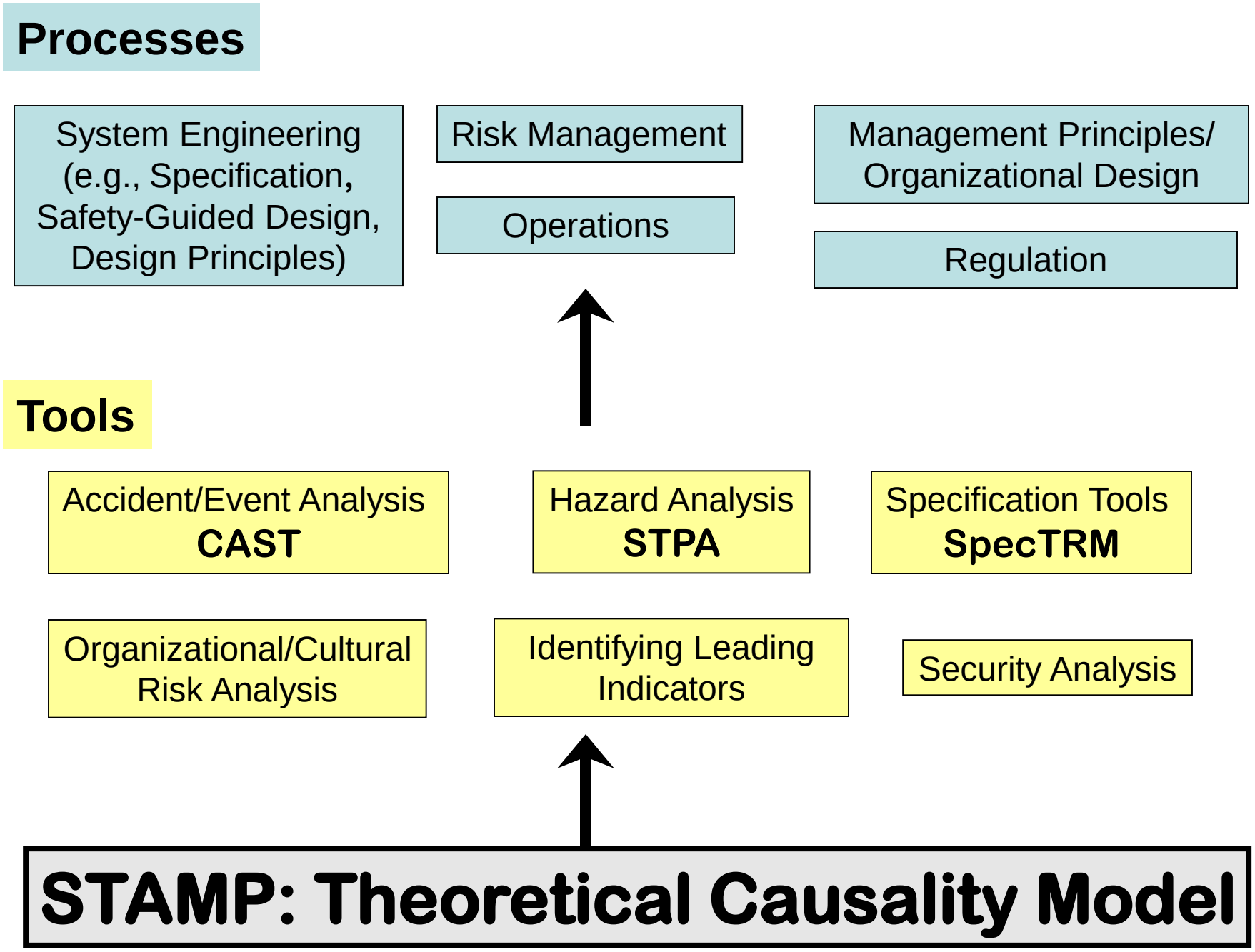
(Leveson, 2003); (Leveson, 2011)

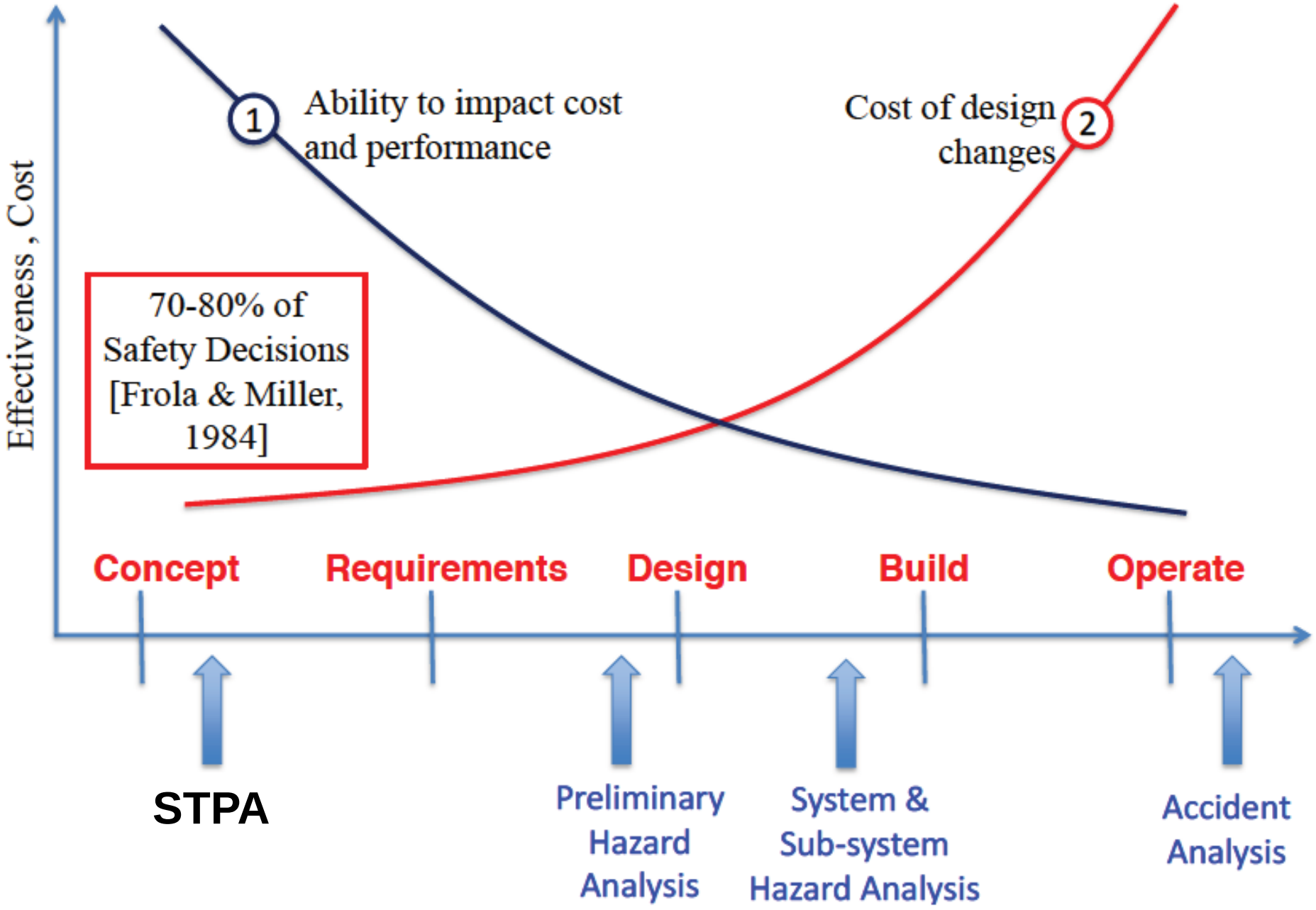
Identifying Causal Scenarios



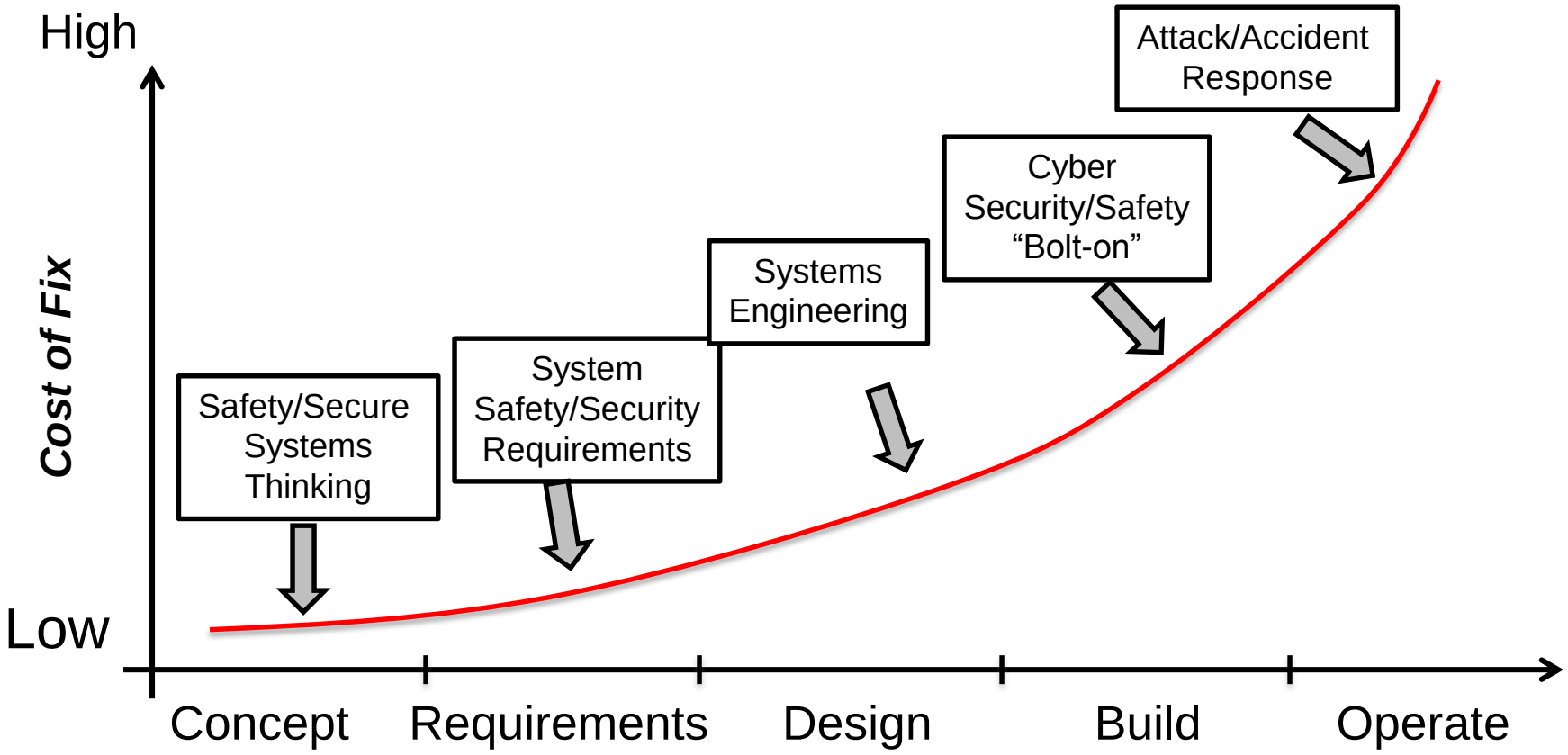
STAMP
(System-Theoretic Accident Model and Processes)

- 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)
- Includes
  - Entire socio-technical system (not just technical part)
  - Component interaction accidents
  - Software and system design errors
  - Human errors


Processes
System Engineering (e.g., Specification, Safety-Guided Design, Design Principles)
Risk Management
Operations
Management Principles/ Organizational Design
Regulation
Tools
Accident/Event Analysis
CAST
Hazard Analysis
STPA
Specification Tools
SpecTRM
Organizational/Cultural Risk Analysis
Identifying Leading Indicators
Security Analysis
STAMP: Theoretical Causality Model


Effectiveness , Cost
1
Ability to impact cost and performance
Cost of design changes
2
70-80% of Safety Decisions [Frola & Miller, 1984]
Concept
Requirements
Design
Build
Operate
STPA
Preliminary Hazard Analysis
System & Sub-system Hazard Analysis
Accident Analysis

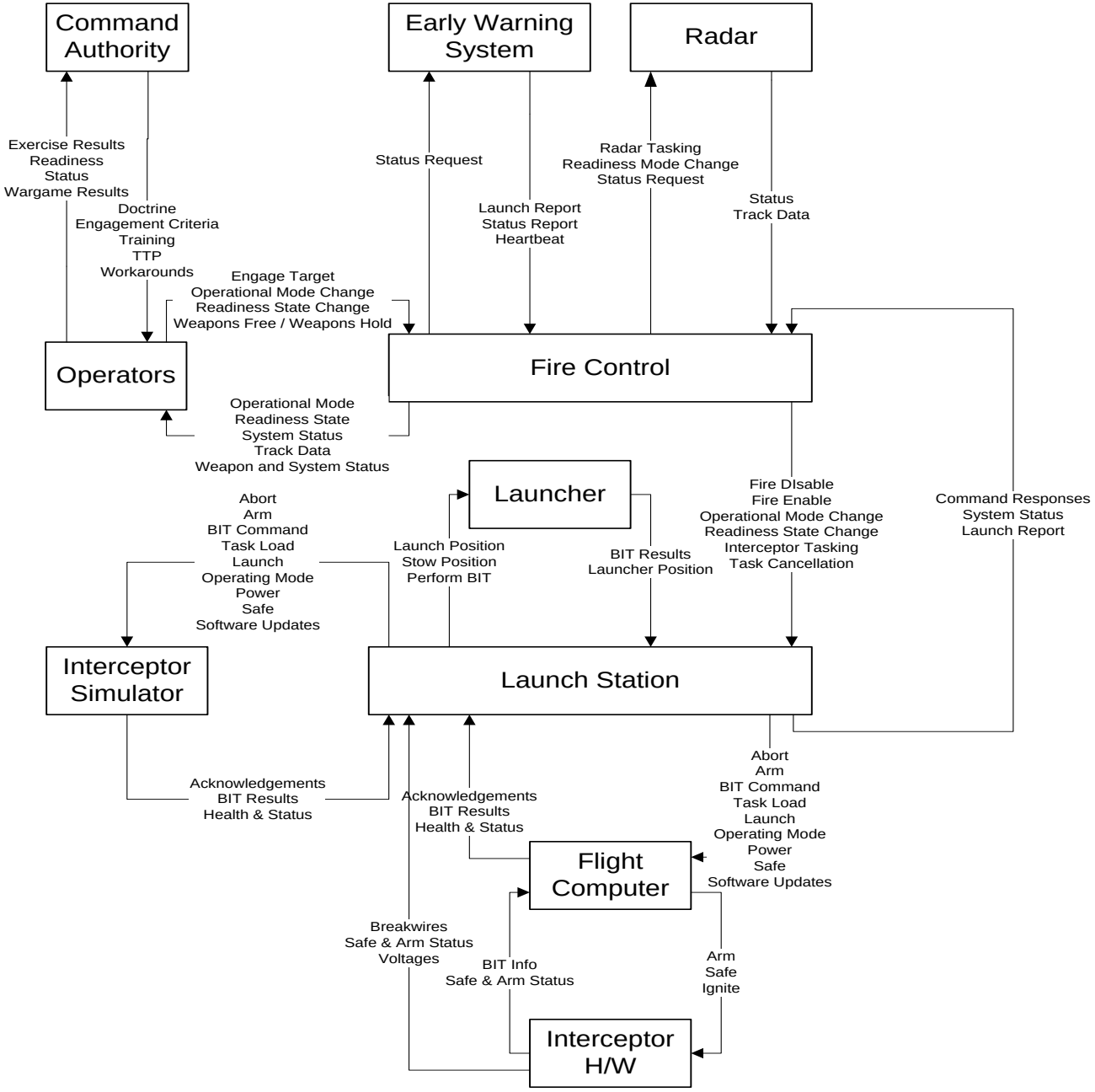
Build safety and security into system from beginning



Example U.S. BDMS (for MDA)

- Non-advocate safety assessment just prior to deployment and field testing
- Hazard was inadvertent launch
- Analysis done by two people over 5 months
- Deployment and testing held up for 6 months because so many scenarios identified for inadvertent launch. In many of these scenarios:
  - All components were operating exactly as intended
  - Complexity of component interactions led to unanticipated system behavior
- STPA also identified component failures that could cause inadequate control (most analysis techniques consider only these failure events)

Safety Control Structure for FMIS



Hazard: Inadvertent/Incorrect Launch

Fire Control Computer

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Wrong Timing/Order Causes Hazard	Stopping too soon
Fire Enable	Not unsafe	When no threat detected	[early/late: same as providing] Discover track spurious so send fire disable but disable arrives before enable	n/a

Example Causes Identified

1. Providing Fire Enable causes hazard
   - Process model incorrectly thinks threat exists
   - The fire control computer is intended to send the fire enable command to the launch station upon receiving a weapons free command from an FMIS operator and while the fire control system has at least one active track
   - The specification requires an “active” track
   - The software supports declaring tracks inactive after a certain period with no radar input, after the total predicted impact time for the track, and/or after a confirmed intercept
   - One case was not well considered: if an operator de-selects all of these options
   - The inadvertent or intentional entry of a weapons free command would send the fire enable command to the launch station even if there were no threats to engage currently tracked by the system

FMIS Inadequate Controls (cont'd)

2. Providing Fire Enable causes hazard
   - Process model thinks in test simulation mode
   - The FMIS system undergoes periodic system operability testing using an interceptor simulator that mimics the interceptor flight computer
   - Traditional hazard analysis of the system identified the possibility that commands intended for test activities could be sent to the operational system
   - System status information provided by the LS includes whether the LS is connected only to missile simulators or to any live interceptors
   - If the fire control computer detects a change in this state, it will warn the operator and offer to reset into a matching state
   - There is a small window of time before the LS notifies the fire control component of the change during which the fire control software might send a fire enable command intended for test to the live LS

FMIS Inadequate Controls (cont'd)

2. Fire Enable arrives after Fire Disable when spurious track detected
   - Fire Enable sent before Fire Disable
   - Discover track is spurious
   - The two commands are sent on separate communication paths
   - Order of arrival could be different than order sent

FHA: SAE ARP 4761
BSCU (Brake System Control Unit)

- The probability of “BSCU Fault Causes Loss of Braking Commands” shall be less than 3.3E-5 per flight.
- The probability of “Loss of a single BSCU shall be less than 5.75E per flight.
- The probability of “Loss of Normal Brake System Hydraulic Components” shall be less than 3.3E-5 per flight.
- The probability of “Inadvertent braking due to BSCU” shall be less than 2.5E-9 per flight.
- No single failure of the BSCU shall lead to “inadvertent braking.”
- The BSCU shall be designed to Development Assurance Level A based on the catastrophic classification of “inadvertent braking due to BSCU”

UNSAFE CONTROL ACTION – BSCU.1a2: Brake command not provided during landing roll, resulting in insufficient deceleration and potential overshoot

Scenario 1: Autobrake believes the desired deceleration rate has already been achieved or exceeded (incorrect process model). The reasons Autobrake may have this process model flaw include:

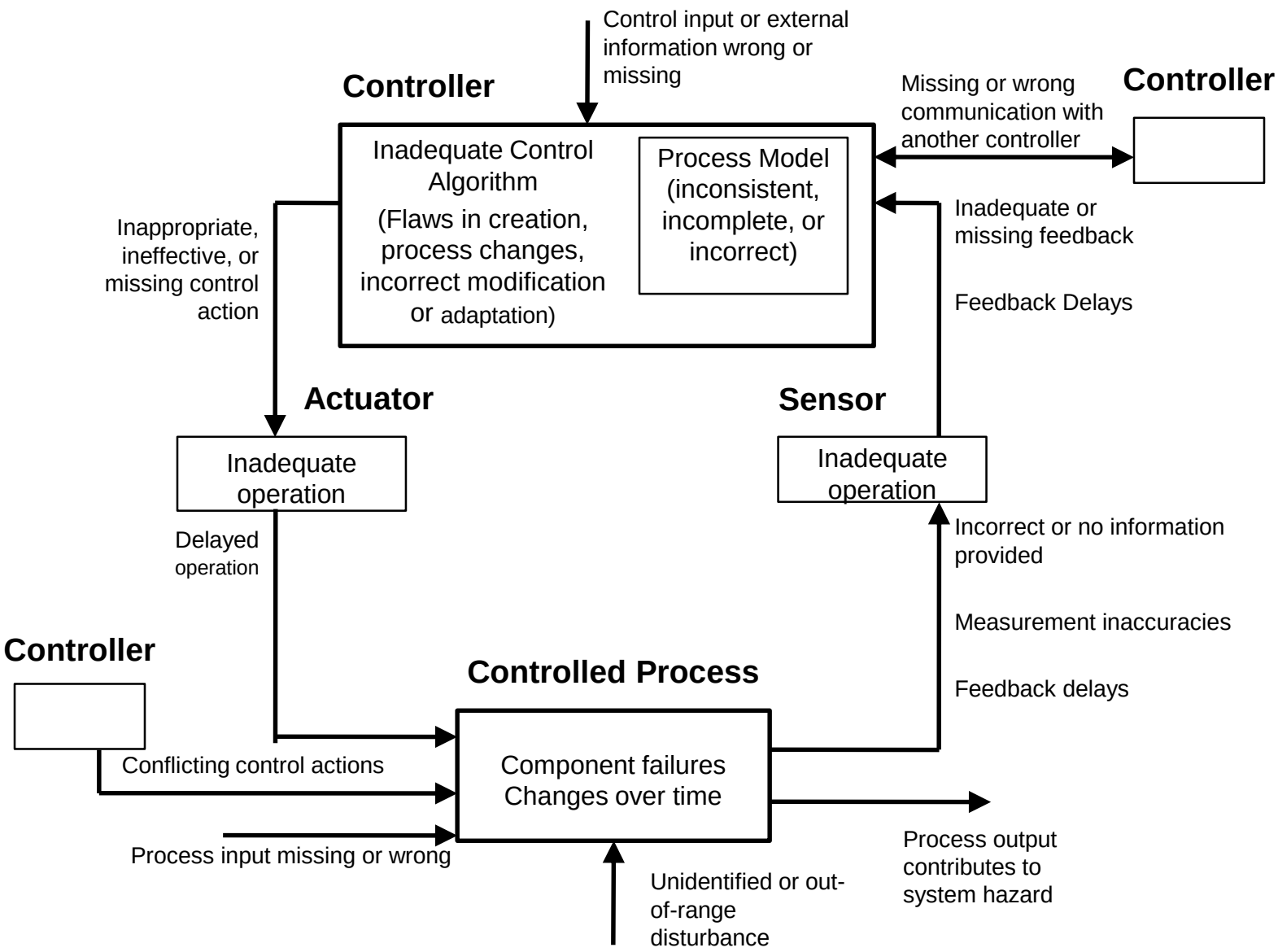
- If wheel speed feedback influences the deceleration rate determined by the Autobrake controller, inadequate wheel speed feedback may cause this scenario. Rapid pulses in the feedback (e.g. wet runway, brakes pulsed by anti-skid) could make the actual aircraft speed difficult to detect and an incorrect aircraft speed might be assumed.
- Inadequate external speed/deceleration feedback could explain the incorrect Autobrake process model (e.g. inertial reference drift, calibration issues, sensor failure, etc.).

Possible Requirement for S1: Provide additional feedback to Autobrake to detect aircraft deceleration rate in the event of wheel slipping (e.g. fusion of multiple sensors)

Integrated Approach to Safety and Security (Col. Bill Young)

- Safety: prevent losses due to unintentional actions by benevolent actors
- Security: prevent losses due to intentional actions by malevolent actors
- Key difference is intent
- Common goal: loss prevention
  - Ensure that critical functions and services provided by networks 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

Identifying Causal Scenarios



Example: Stuxnet

- Loss: Damage to reactor (in this case centrifuges)
- Hazard/Vulnerability: Centrifuges are damaged by spinning too fast
- Constraint: 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

Evaluation: Does it Work?

Is it Practical?

- STPA has been or is being used in a large variety of industries
  - Spacecraft
  - Aircraft
  - Air Traffic Control
  - UAVs (RPAs)
  - Defense
  - Automobiles
  - Medical Devices and Hospital Safety
  - Chemical plants
  - Oil and Gas
  - Nuclear and Electrical Power
  - Finance
  - Etc.

Is it Effective?

- Most of these systems are very complex (e.g., the new U.S. missile defense system)
- In all cases where a comparison was made (to FTA, HAZOP, FMEA, ETA, etc.)
  - STPA found the same hazard causes as the old methods
  - Plus it found more causes than traditional methods
  - In some evaluations, found accidents that had occurred that other methods missed (e.g., EPRI)
  - Cost was orders of magnitude less than the traditional hazard analysis methods
  - Same results for security evaluations by CYBERCOM

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

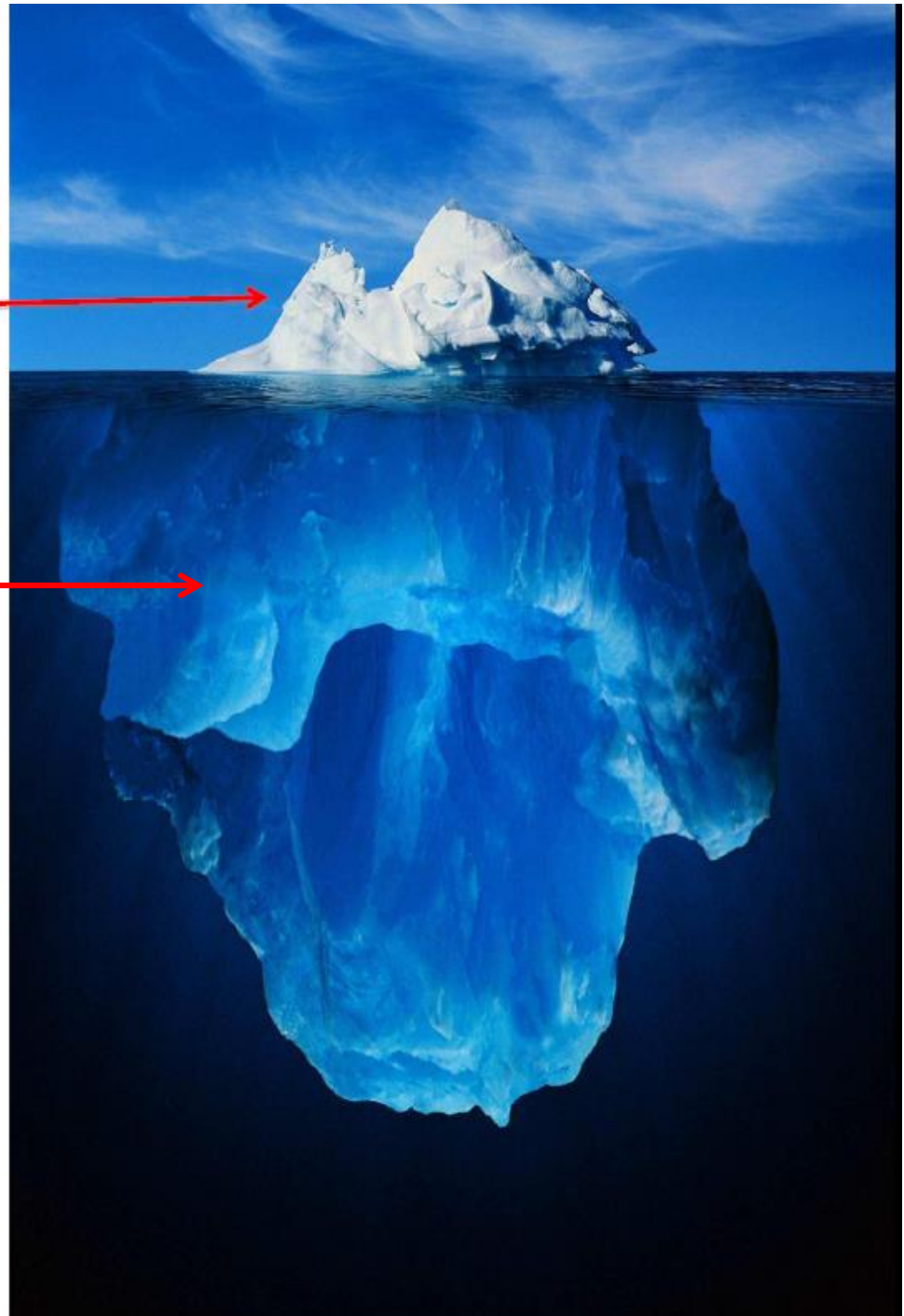
Value of new paradigm often depends on ability to accommodate successes and empirical observations made in old paradigm.

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

LEVERAGE

Event Based Models and Methods

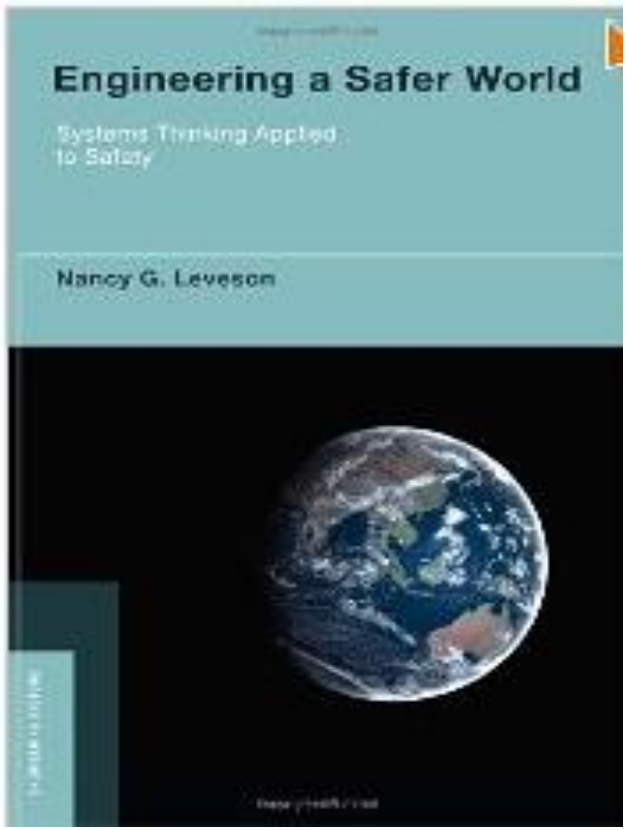
STAMP and STPA



WYOUNG@MIT.EDU

Nancy Leveson, *Engineering a Safer World:*

Systems Thinking Applied to Safety



MIT Press, January 2012

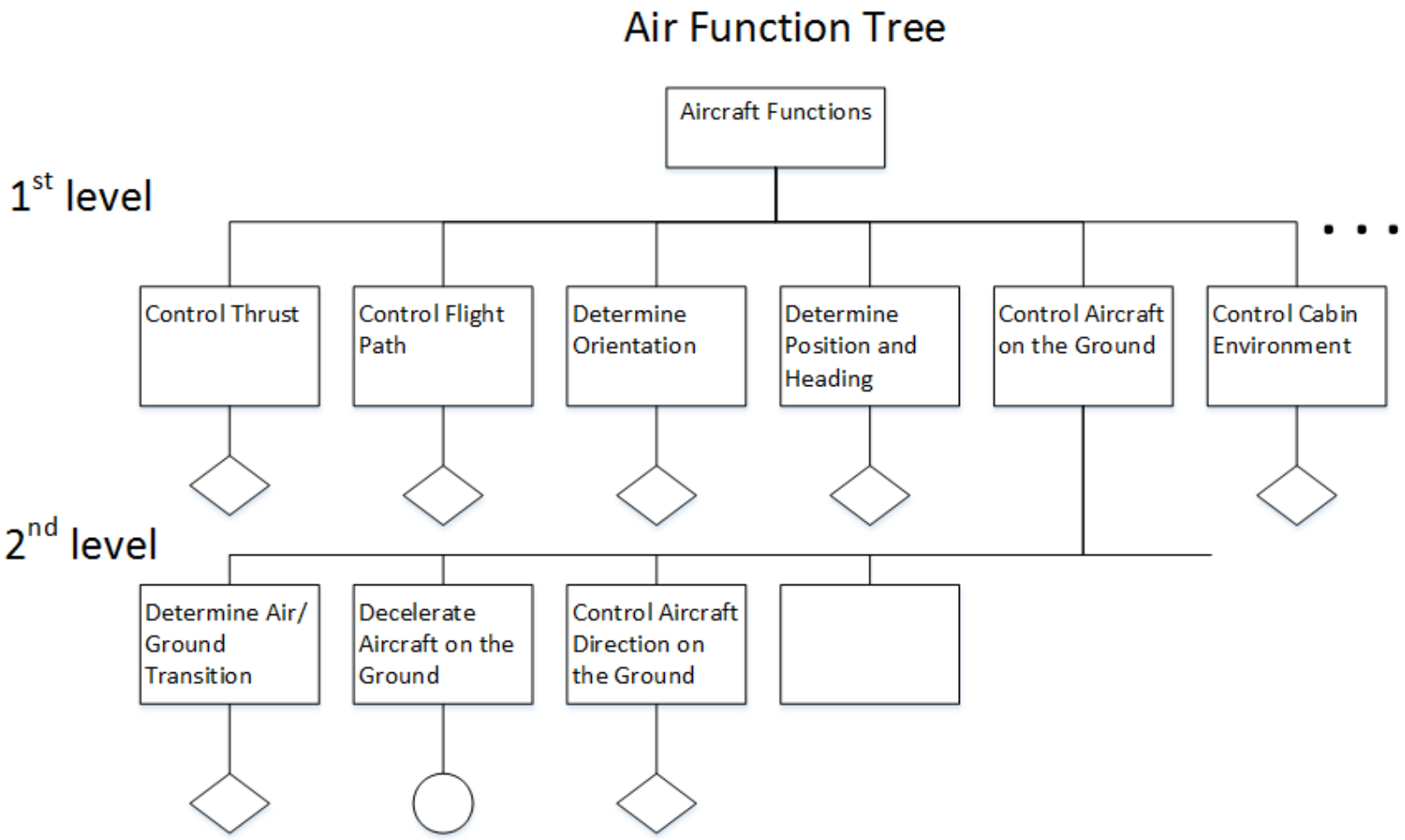
MIT STAMP Workshop 2017

- March 27-30, 2017
- Tutorials and presentations
- Usually 200-300 people attend (from 20-25 countries)

Information at:

http://psas.scripts.mit.edu/home/

SAE ARP 4761: Commercial Aircraft Certification



ARP 4761

Function	Failure Condition (Hazard Description)	Phase	Effect of Failure Condition on Aircraft/Crew	Classification
Decelerate Aircraft on the Ground	a. Unannunciated loss of deceleration capability	Landing/ RTO	Crew is unable to decelerate the aircraft resulting in a high speed overrun	Catastrophic
	b. Annunciated loss of deceleration capability	Landing	Crew selects a more suitable airport, notifies emergency ground support and prepares occupants for landing overrun.	Hazardous
	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
	Inadvertent Deceleration after Vl (Takeoff/RTO decision speed)	Takeoff	Crew is unable to takeoff due to application of brakes at the same time as high thrust settings resulting in a high speed overrun	Catastrophic

Example Aircraft Level Requirements Generated

- Loss of all wheel braking during landing or rejected takeoff (RTO) shall be less than 5E-7 per flight
- Asymmetrical loss of wheel braking coupled with loss of rudder or nose wheel steering during landing or RTO shall be less than 5E-7 per flight
- Inadvertent wheel braking with all wheels locked during takeoff roll before V1 shall be less than 5E-7 per flight.
- Inadvertent wheel braking of all wheels during takeoff roll after V1 shall be less than 5E-9 per flight.
- Undetected inadvertent wheel braking on one wheel without locking during takeoff shall be less than 5E-9 per flight.

FHA: SAE ARP 4761
BSCU (Brake System Control Unit)

- The probability of “BSCU Fault Causes Loss of Braking Commands” shall be less than 3.3E-5 per flight.
- The probability of “Loss of a single BSCU shall be less than 5.75E per flight.
- The probability of “Loss of Normal Brake System Hydraulic Components” shall be less than 3.3E-5 per flight.
- The probability of “Inadvertent braking due to BSCU” shall be less than 2.5E-9 per flight.
- No single failure of the BSCU shall lead to “inadvertent braking.”
- The BSCU shall be designed to Development Assurance Level A based on the catastrophic classification of “inadvertent braking due to BSCU”

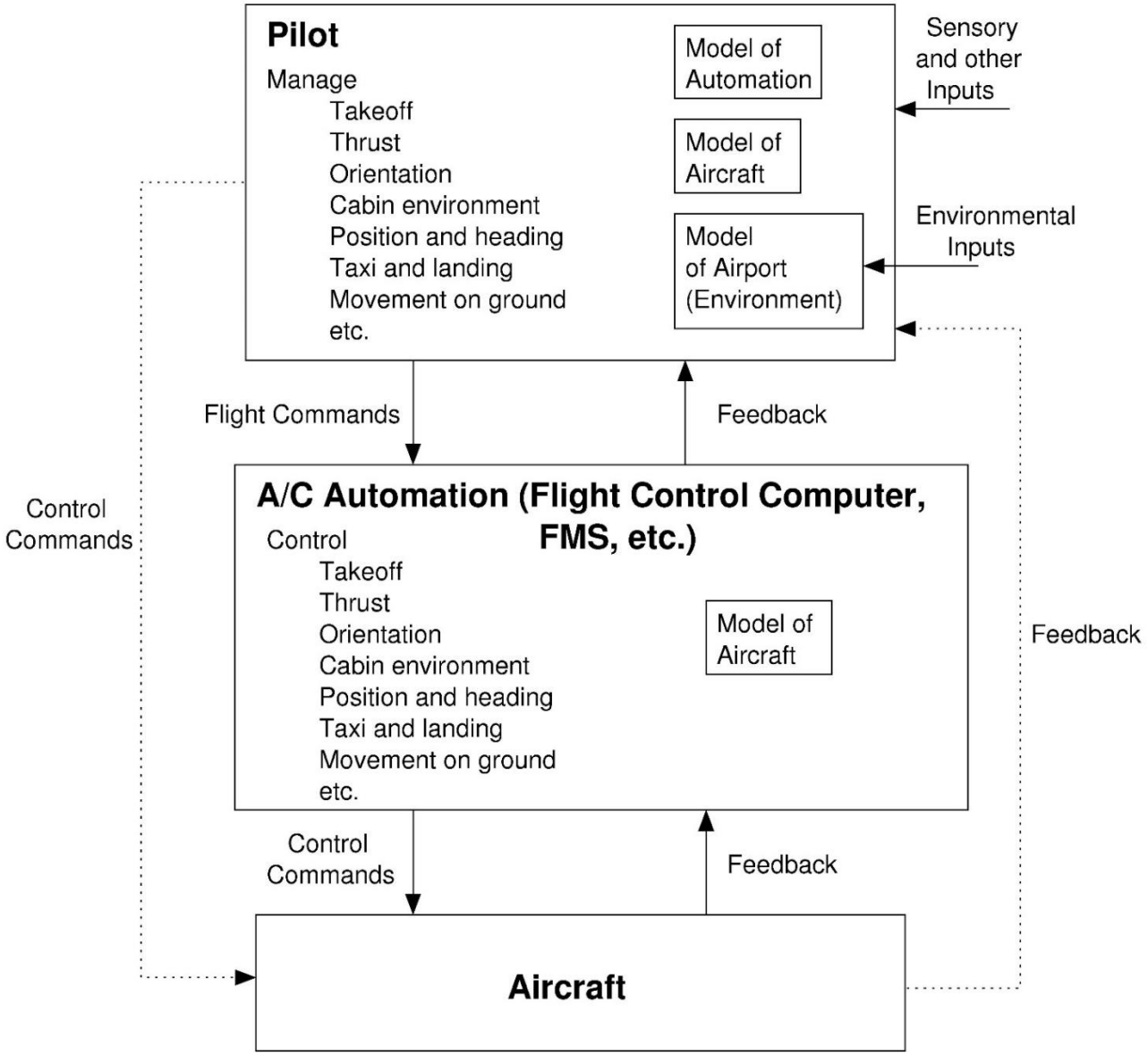
STPA: System-Theoretic Process Analysis

- A top-down, system engineering analysis technique
- Identifies safety (or X) constraints (system and component requirements)
- Identifies scenarios leading to violation of constraints (requirements); use results to design or redesign system to be safer
- Can be used on technical design and organizational design
- Supports a safety-driven design process where
  - Analysis influences and shapes early design decisions
  - Analysis iterated and refined as design evolves

STPA: Systems Theoretic Process Analysis

- Works on the control structure
  1. Identify the potential unsafe control actions
  2. Identify scenarios (causes) that can lead to them
  3. Design the system to eliminate or control them
- A structured step-by-step process that can be partially automated

STPA Control Structure Model



UNSAFE CONTROL ACTION – CREW.1a1: Crew does not provide manual braking when there is no Autobraking and braking is necessary to prevent H4-1 and H4-5.

Scenario 1: Crew incorrectly believes that the Autobrake is armed and expect the Autobrake to engage (process model flaw)

Reasons that their process model could be flawed include:

- The crew previously armed Autobrake and does not know it later became unavailable

AND/OR

- The feedback received is adequate when the BSCU Hydraulic Controller detects a fault. The crew would be notified of a generic BSCU fault but they are not notified that Autobrake is still armed (even though Autobraking is no longer available)

AND/OR

- The crew is notified that the Autobrake controller is still armed and ready, because the Autobrake controller does not detect when the BSCU has detected a fault. When the BSCU detects a fault it closes the green shut-off valve (making Autobrake commands ineffective), but the Autobrake system itself does not notify the crew.
- The crew cannot process feedback due to multiple messages, conflicting messages, alarm fatigue, etc.

Possible new requirements for S1: The BSCU hydraulic controller must provide feedback to the Autobrake when it is faulted and the Autobrake must disengage (and provide feedback to crew).

Other requirements may be generated from a human factors analysis of the ability of the crew to process the feedback under various worst-case conditions.

UNSAFE CONTROL ACTION – BSCU.1a2: Brake command not provided during landing roll, resulting in insufficient deceleration and potential overshoot

Scenario 1: Autobrake believes the desired deceleration rate has already been achieved or exceeded (incorrect process model). The reasons Autobrake may have this process model flaw include:

- If wheel speed feedback influences the deceleration rate determined by the Autobrake controller, inadequate wheel speed feedback may cause this scenario. Rapid pulses in the feedback (e.g. wet runway, brakes pulsed by anti-skid) could make the actual aircraft speed difficult to detect and an incorrect aircraft speed might be assumed.
- Inadequate external speed/deceleration feedback could explain the incorrect Autobrake process model (e.g. inertial reference drift, calibration issues, sensor failure, etc.).

Possible Requirement for S1: Provide additional feedback to Autobrake to detect aircraft deceleration rate in the event of wheel slipping (e.g. fusion of multiple sensors)