Week 2 Class Notes

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Plan for Today

- Accident Models
- Introduction to Systems Thinking
- STAMP: A new loss causality model

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

Traditional Ways to Cope with Complexity

1. Analytic Reduction
2. Statistics

Analytic Reduction

- Divide system into distinct parts for analysis

  Physical aspects → Separate physical components or functions

  Behavior → Events over time

- Examine parts separately and later combine analysis results

- Assumes such separation does not distort phenomenon
  - Each component or subsystem operates independently
  - Analysis results not distorted when consider components separately
  - Components act the same when examined singly as when playing their part in the whole
  - Events not subject to feedback loops and non-linear interactions

Standard Approach to Safety

- Reductionist
  - Divide system into components
  - Assume accidents are caused by component failure
  - Identify chains of directly related physical or logical component failures that can lead to a loss
  - Assume randomness in the failure events so can derive probabilities for a loss
- Forms the basis for most safety engineering and reliability engineering analysis and design

  Redundancy and barriers (to prevent failure propagation),

  high component integrity and overdesign, fail-safe design, ….

Domino "Chain of events" Model

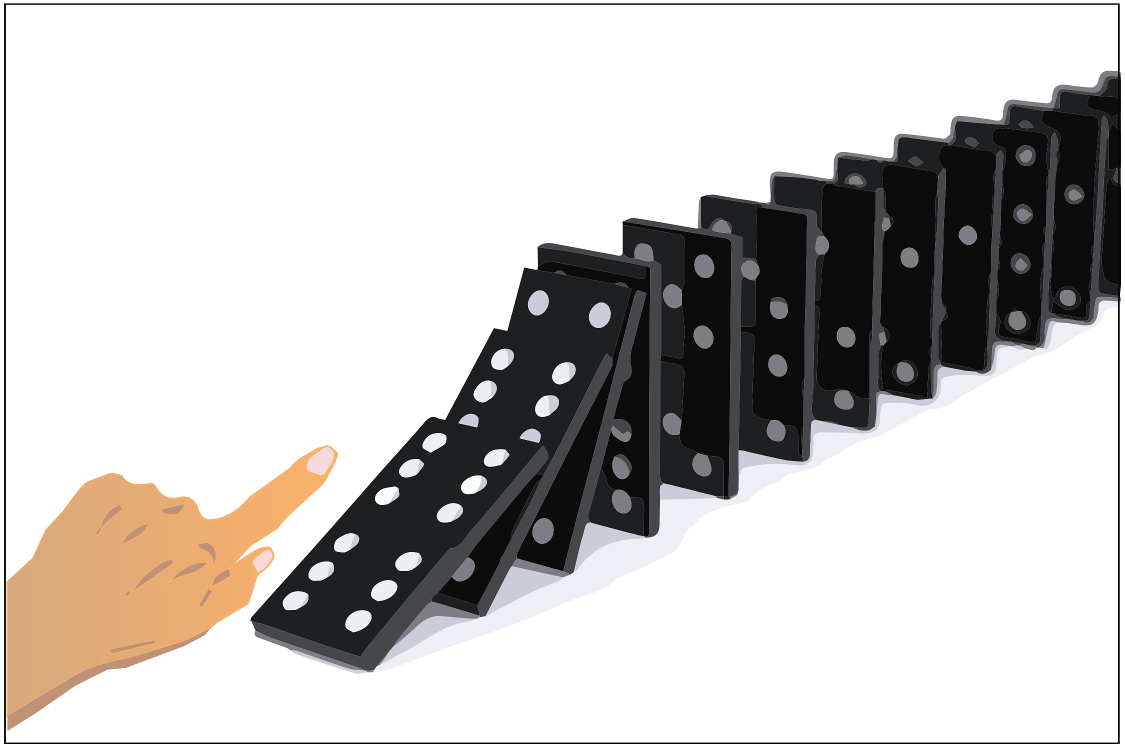


Image by MIT OpenCourseWare.

DC-10:

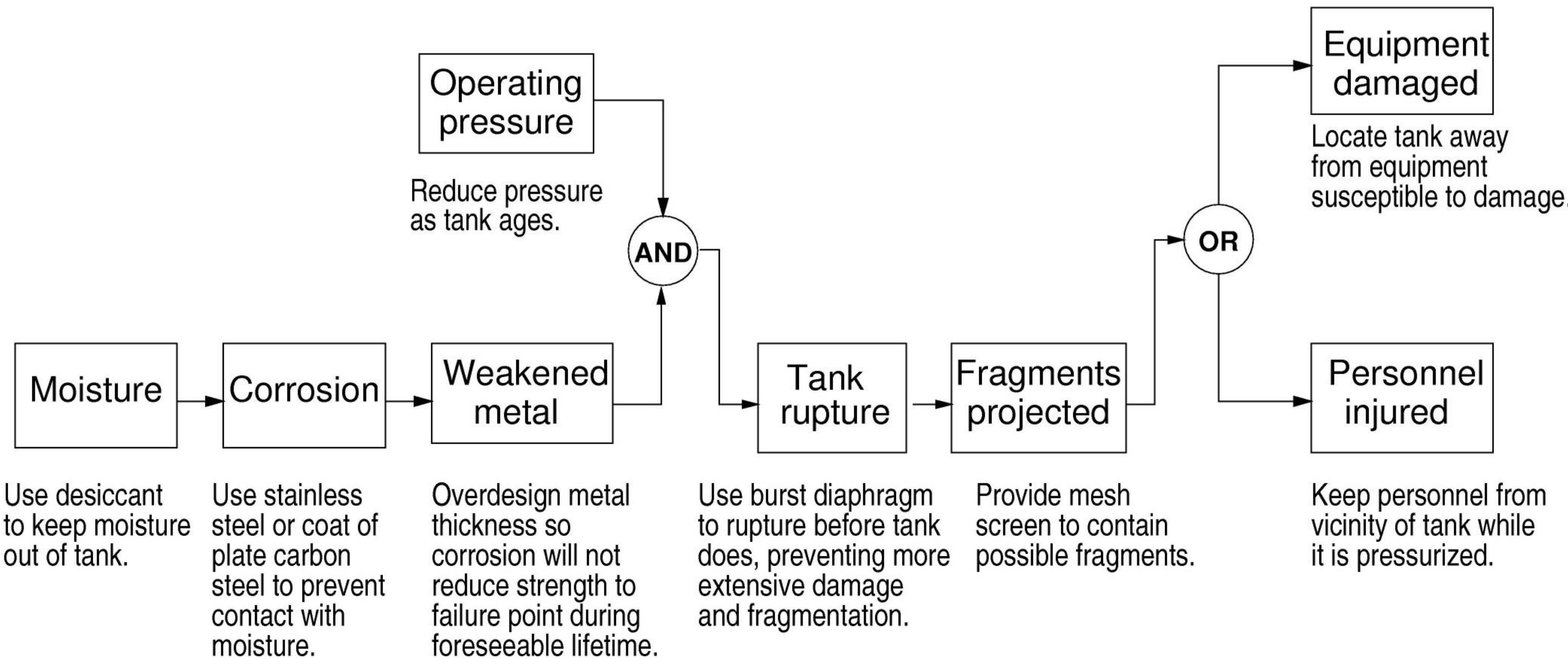
Cargo door fails → (Causes) → Floor collapses → (Causes) → Hydraulics fail → (Causes) → Airplane crashes

Event-based

The Domino Model in action

Image removed due to copyright restrictions.

Chain-of-events example



From Leveson, Nancy (2012). Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, © Massachusetts Institute of Technology. Used with permission.

Event Chain

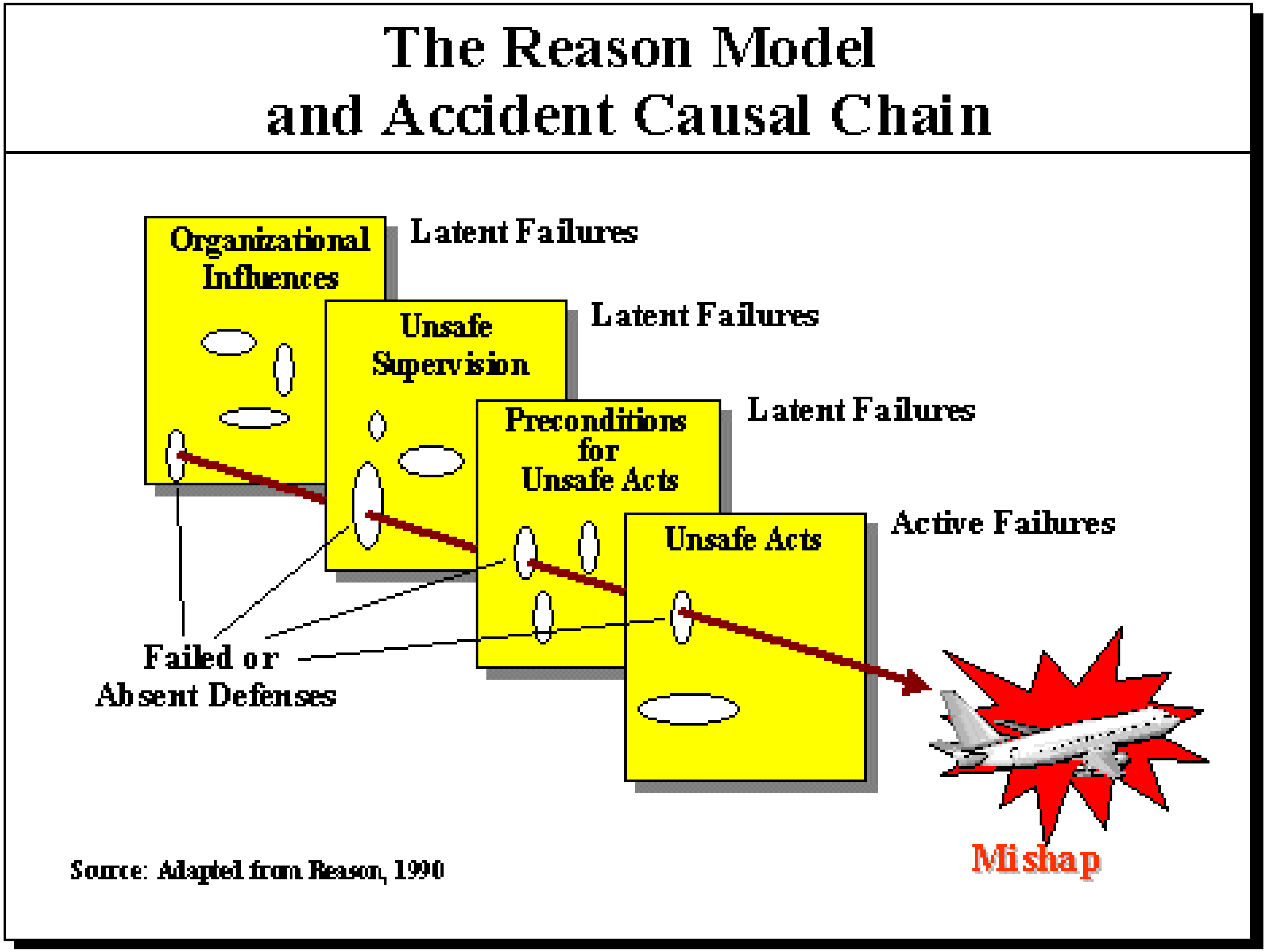
- E1: Worker washes pipes without inserting a slip blind.
- E2: Water leaks into MIC tank
- E3: Gauges do not work
- E4: Operator does not open valve to relief tank
- E3: Explosion occurs
- E4: Relief valve opens
- E5: Flare tower, vent scrubber, water curtain do not work
- E5: MIC vented into air
- E6: Wind carries MIC into populated area around plant.

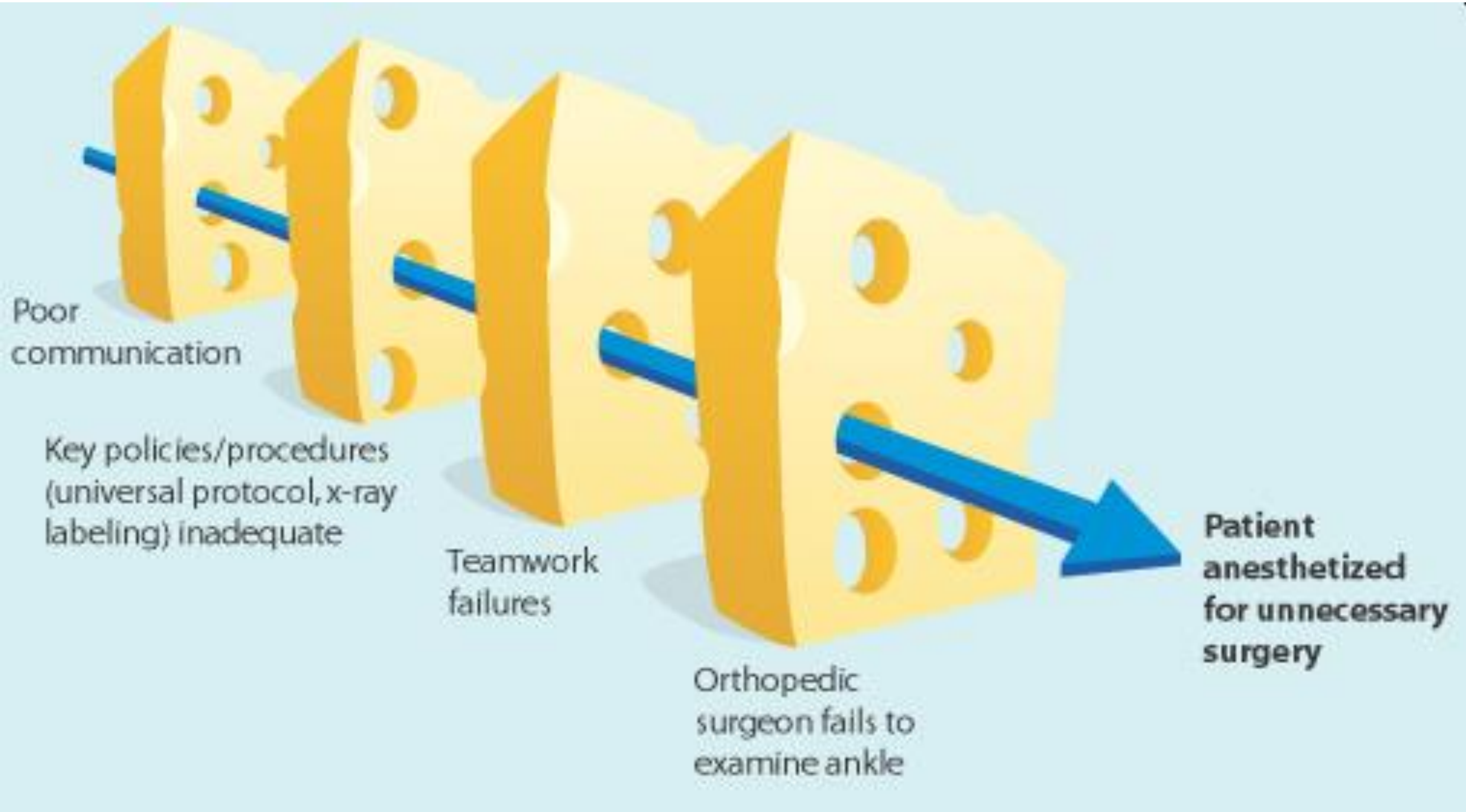
What was the “root cause”?

Variants of Domino Model

- Bird and Loftus (1976)
  - Lack of control by management, permitting
  - Basic causes (personal and job factors) that lead to
  - Immediate causes (substandard practices/conditions/errors), which are the proximate cause of
  - An accident or incident, which results in
  - A loss.
- Adams (1976)
  - Management structure (objectives, organization, and operations)
  - Operational errors (management or supervisor behavior)
  - Tactical errors (caused by employee behavior and work conditions)
  - Accident or incident
  - Injury or damage to persons or property.

Reason Swiss Cheese




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

Swiss Cheese Model Limitations

- Ignores common cause failures of defenses (systemic accident factors)
- Does not include migration to states of high risk
- Assumes accidents are random events coming together accidentally
- Assumes some (linear) causality or precedence in the cheese slices (and holes)
- Just a chain of events, no explanation of “why” events occurred

Accident with No Component Failures

- Mars Polar Lander
  - Have to slow down spacecraft to land safely
  - Use Martian gravity, 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
  - Software not supposed to be operating at that time but software engineers decided to start early to even out load on processor
  - Software thought spacecraft had landed and shut down descent engines

Types of Accidents

- <u>Component Failure Accidents</u>
  - Single or multiple component failures
  - Usually assume random failure
- <u>Component Interaction Accidents</u>
  - Arise in interactions among components
  - Related to interactive and dynamic complexity
  - Behavior can no longer be
    - Planned
    - Understood
    - Anticipated
    - Guarded against
  - Exacerbated by introduction of computers and software

Accident with No Component Failure

- 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.

Analytic Reduction does not Handle

- Component interaction accidents
- Systemic factors (affecting all components and barriers)
- Software and software requirements errors
- Human behavior (in a non-superficial way)
- System design errors
- Indirect or non-linear interactions and complexity
- Migration of systems toward greater risk over time (e.g., in search for greater efficiency and productivity)

Summary

- New levels of complexity, software, human factors do not fit into a reductionist, reliability-oriented world.
- Trying to shoehorn new technology and new levels of complexity into old methods will not work

- “But the world is too complex to look at the whole, we need analytic reduction”
- Right?

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
- Developed for biology and engineering
- First used on ICBM systems of 1950s/1960s

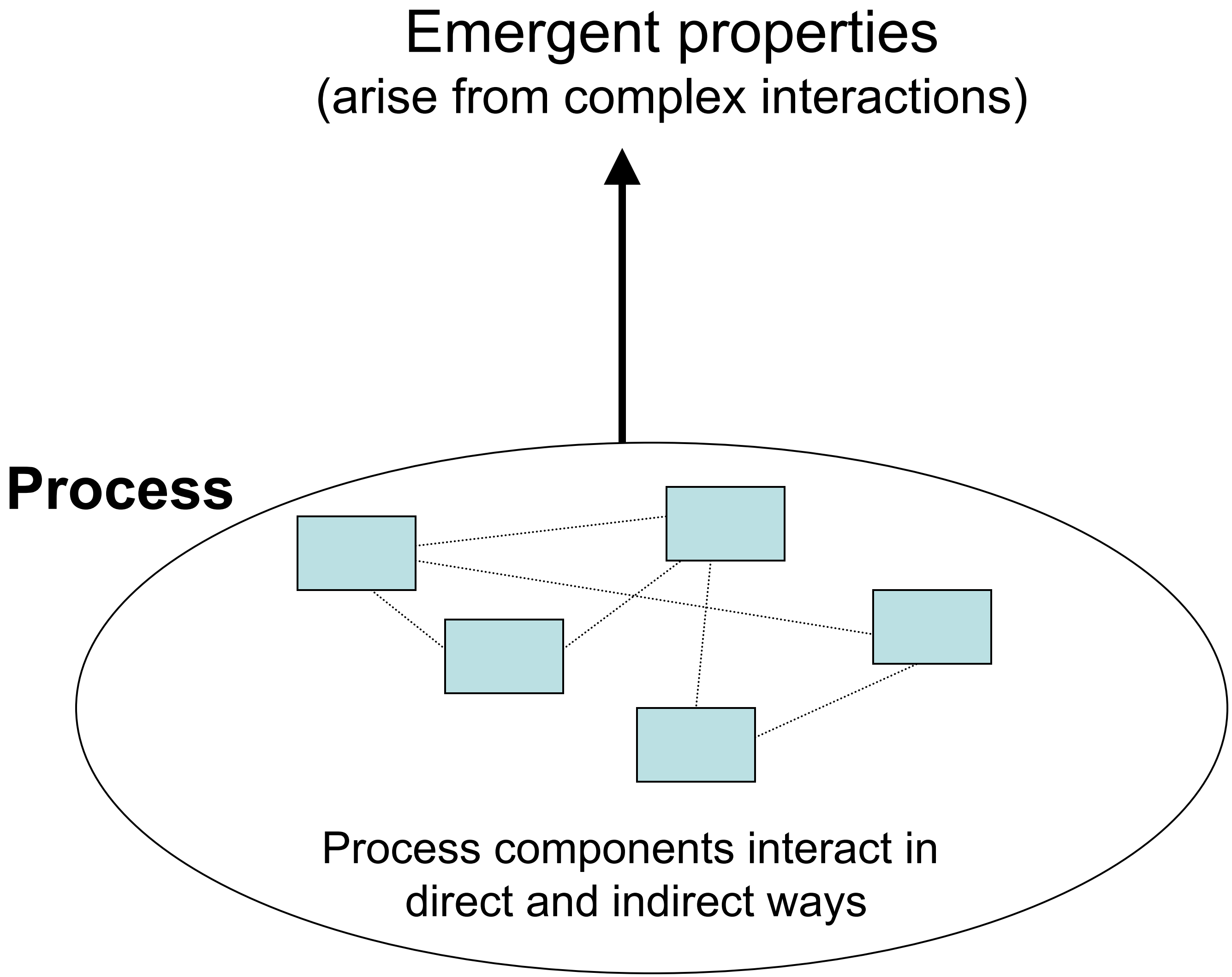
Systems Theory (2)

- 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

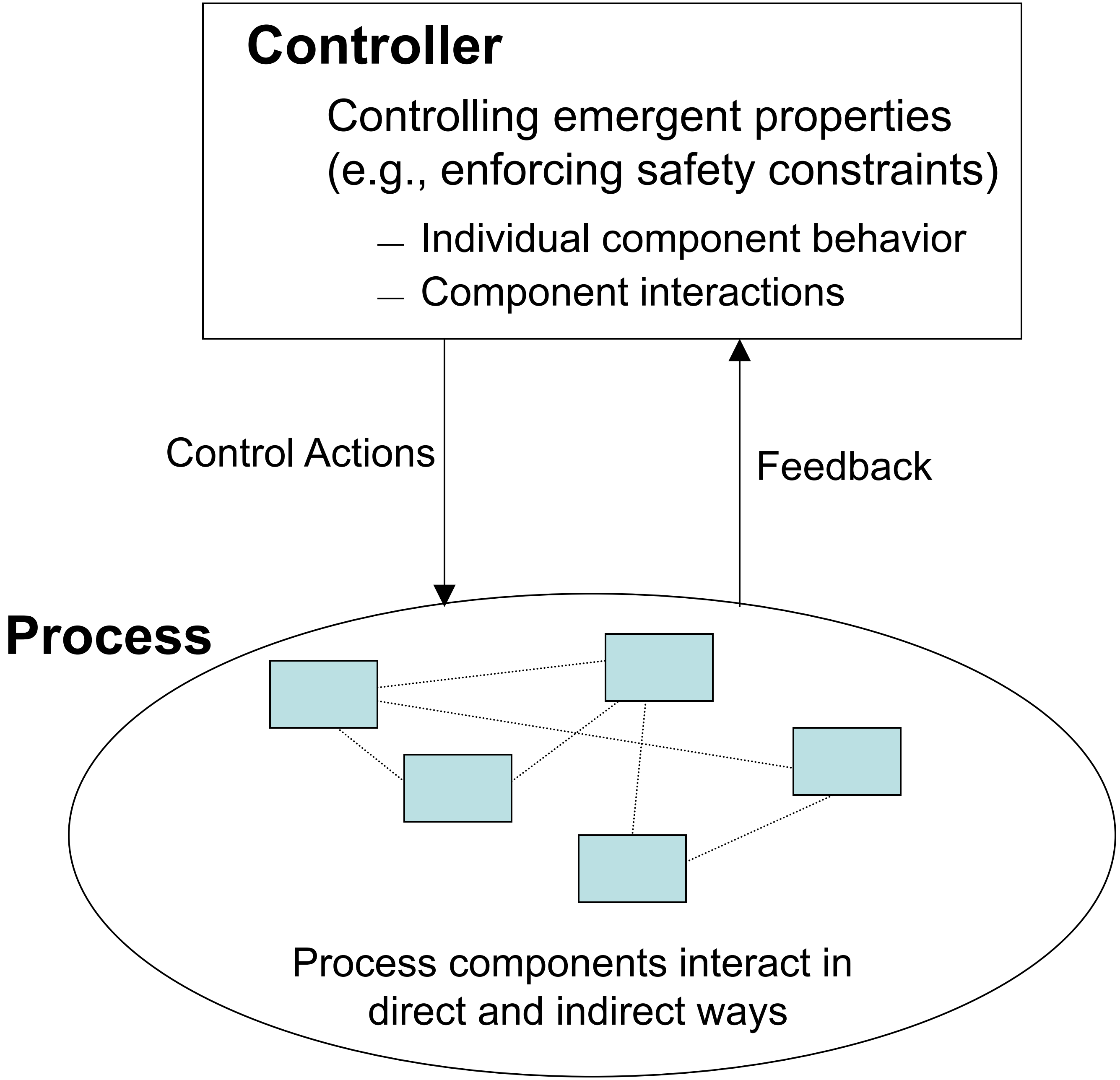
    "The whole is greater than the sum of the parts"

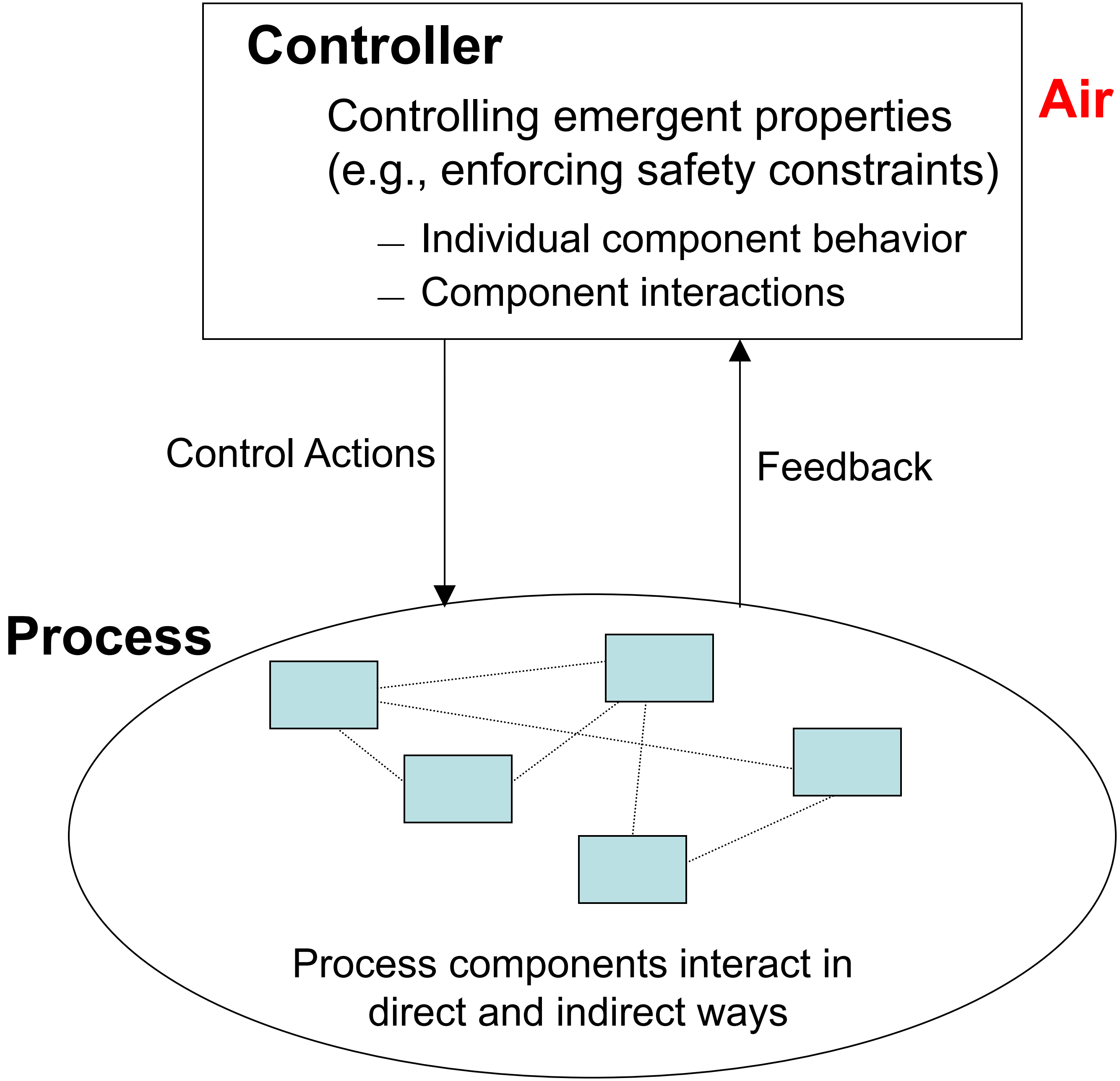
  - These properties arise from relationships among the parts of the system

    How they interact and fit together



Safety is an emergent property


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

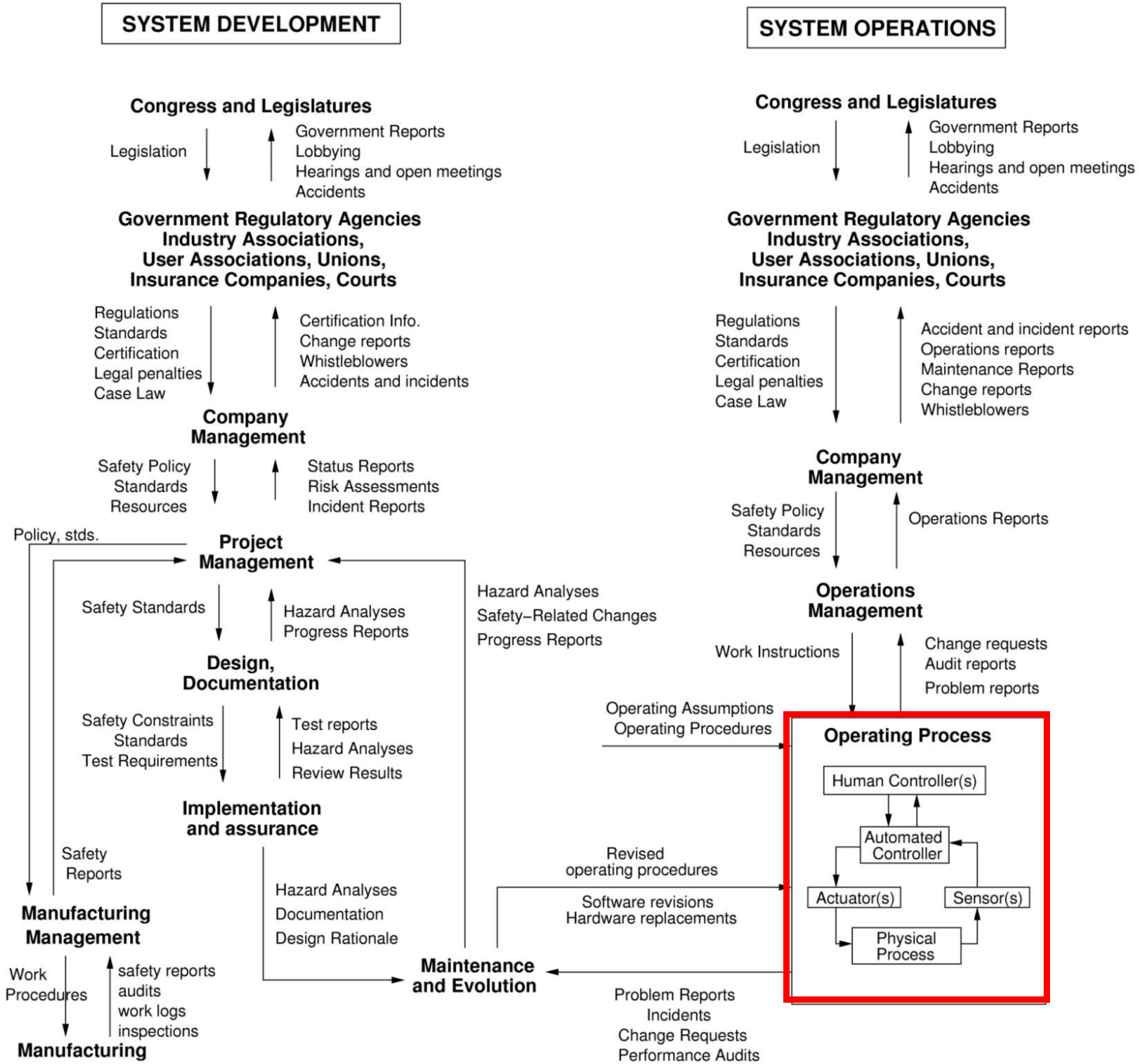


Air Traffic Control:
- **Safety**
- **Throughput**

Controls/Controllers Enforce Safety Constraints

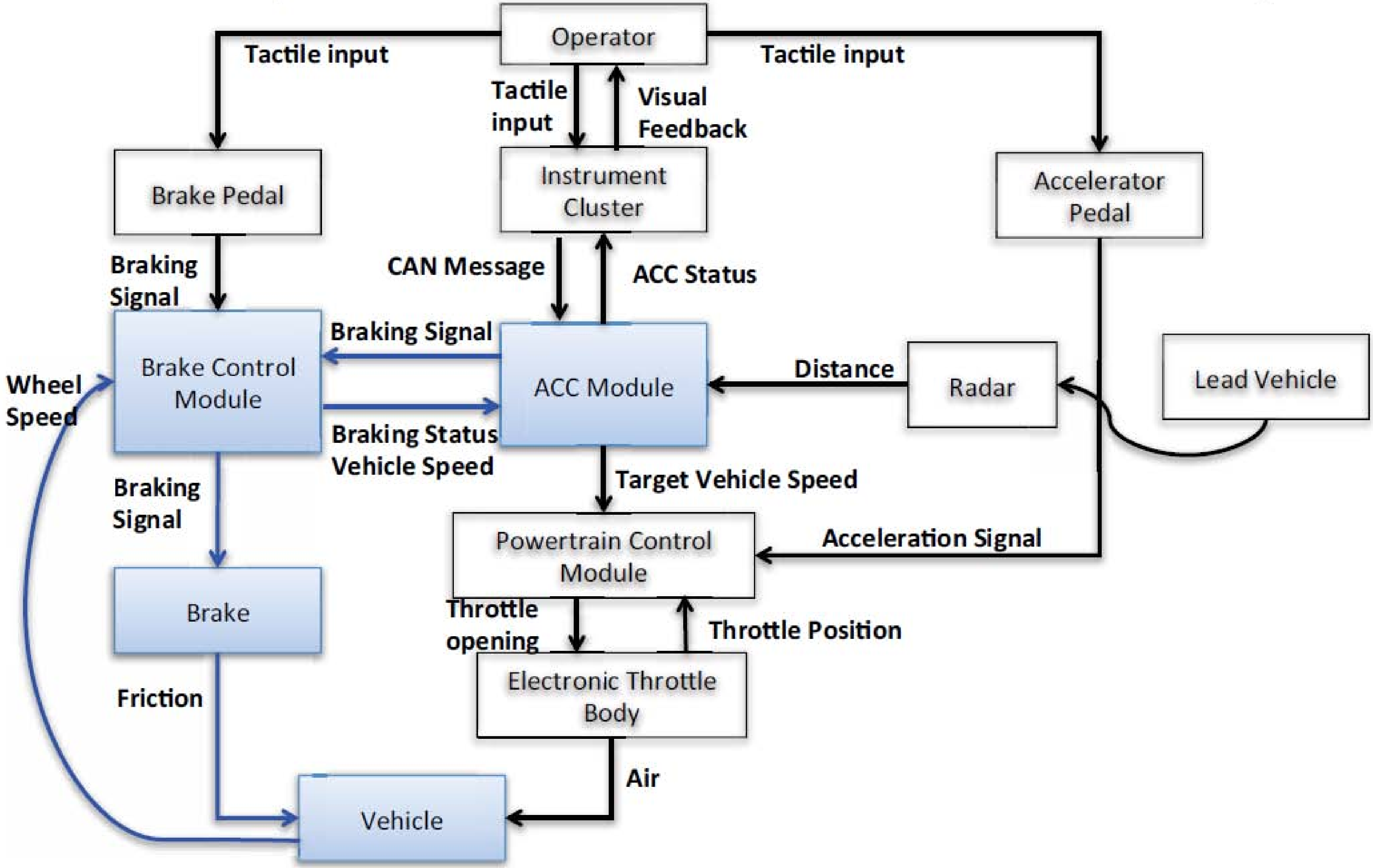
- Power must never be on when access door open
- Two aircraft must not violate minimum separation
- Aircraft must maintain sufficient lift to remain airborne
- Public health system must prevent exposure of public to contaminated water and food products
- Pressure in a deep water well must be controlled
- Truck drivers must not drive when sleep deprived

Example Safety Control Structure

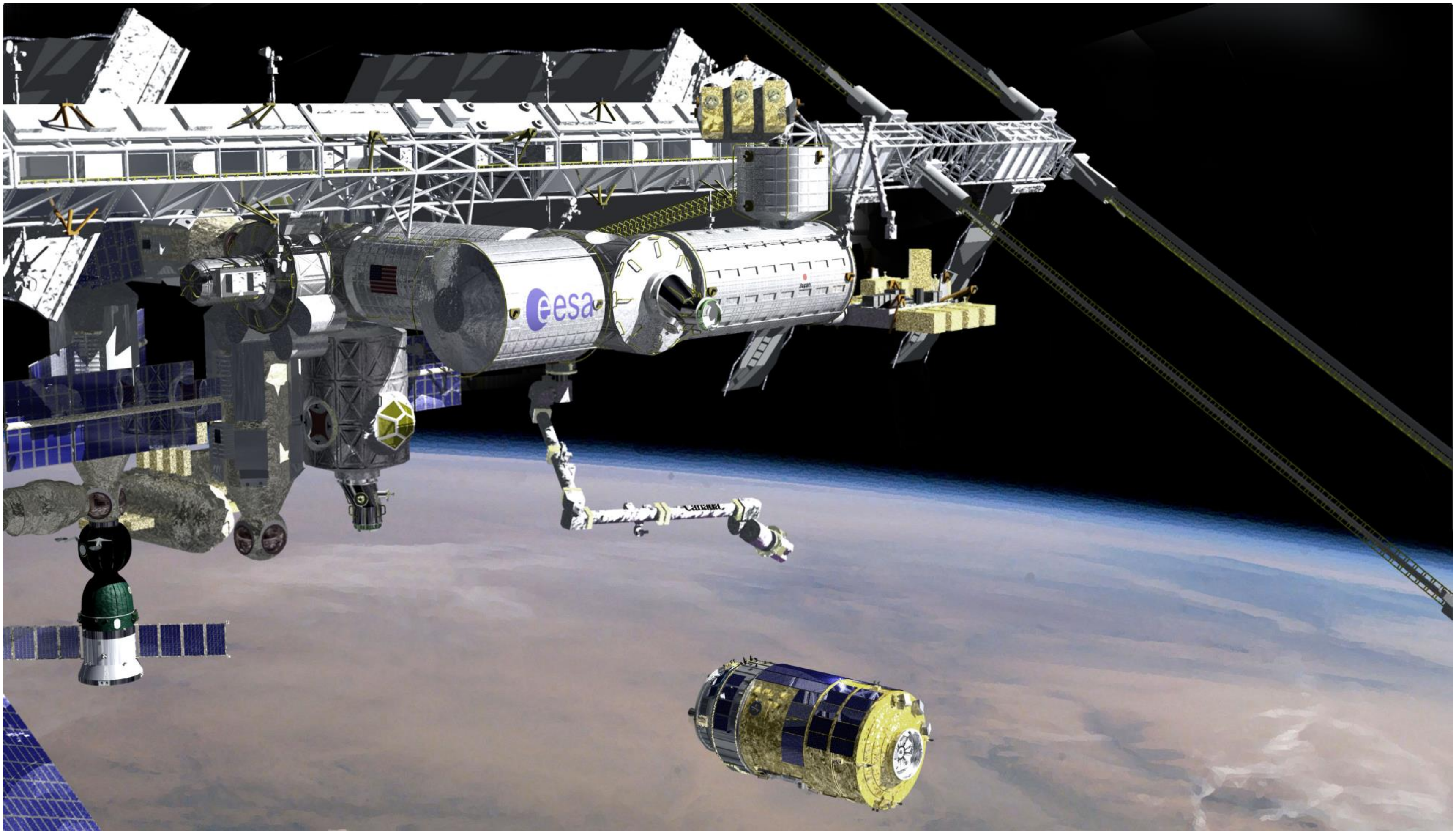


From Leveson, Nancy (2012). Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, © Massachusetts Institute of Technology. Used with permission.

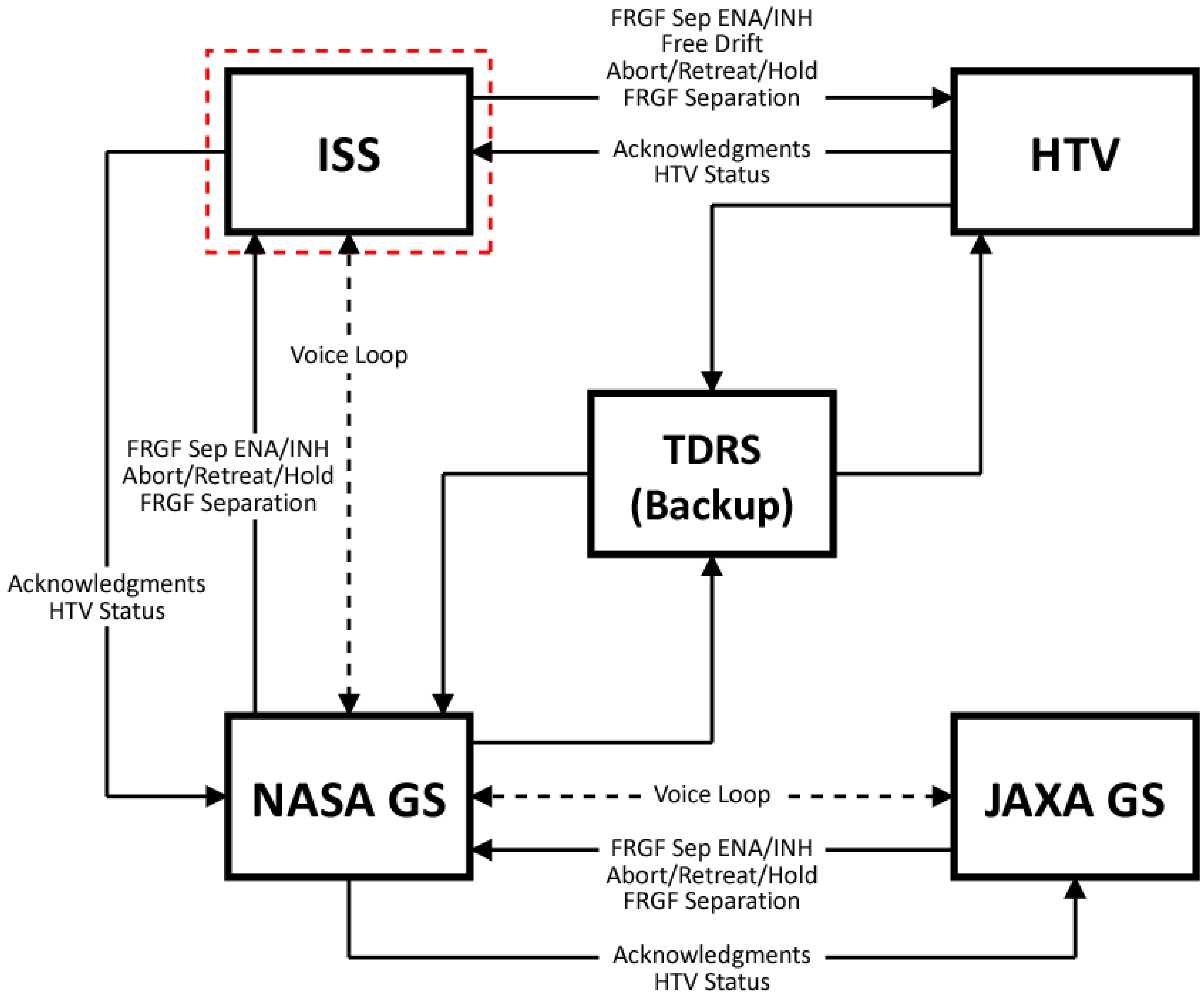
Example: ACC – BCM Control Loop



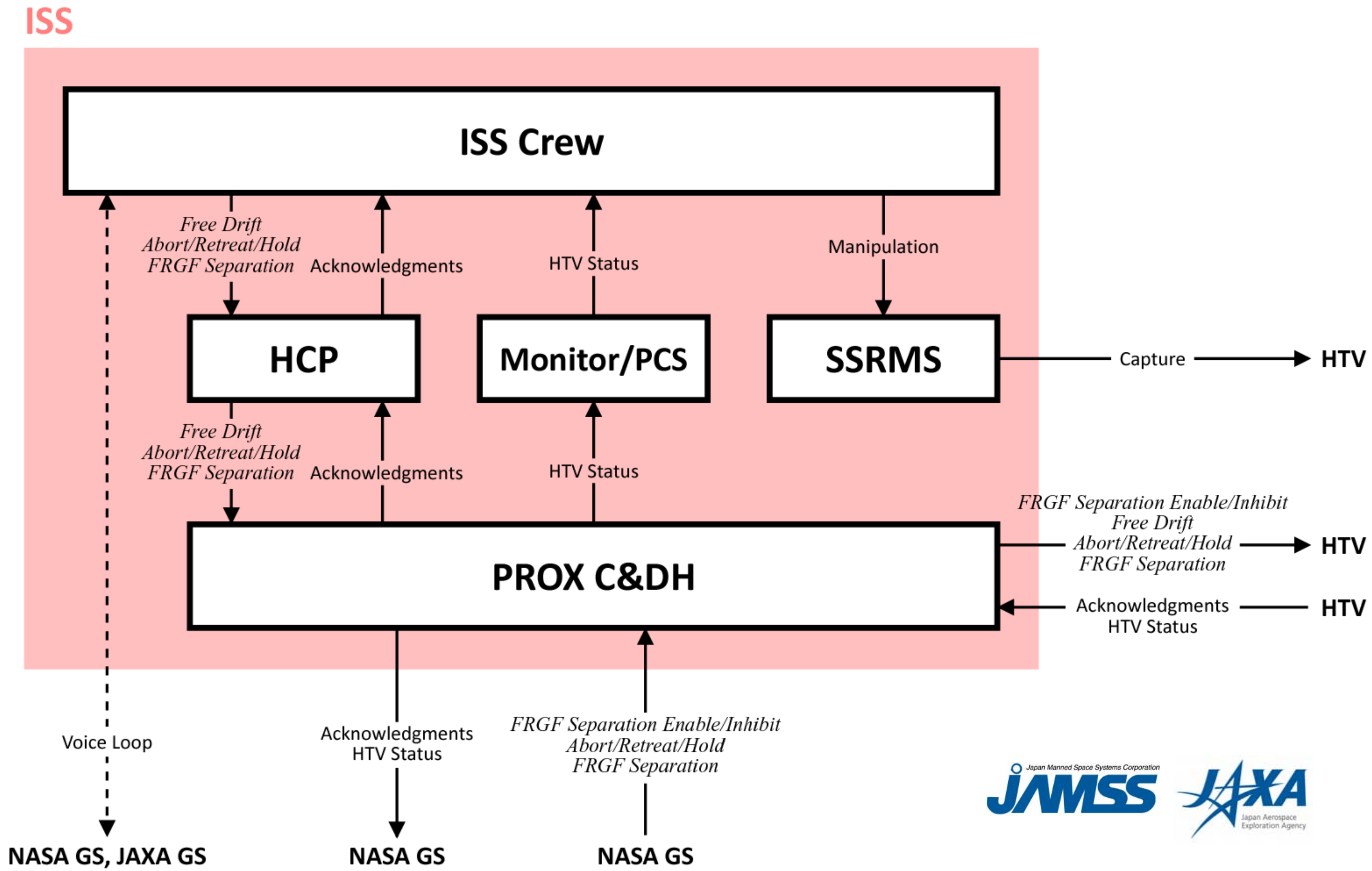
Courtesy of Qi D. Van Eikema Hommes. Used with permission.


esa

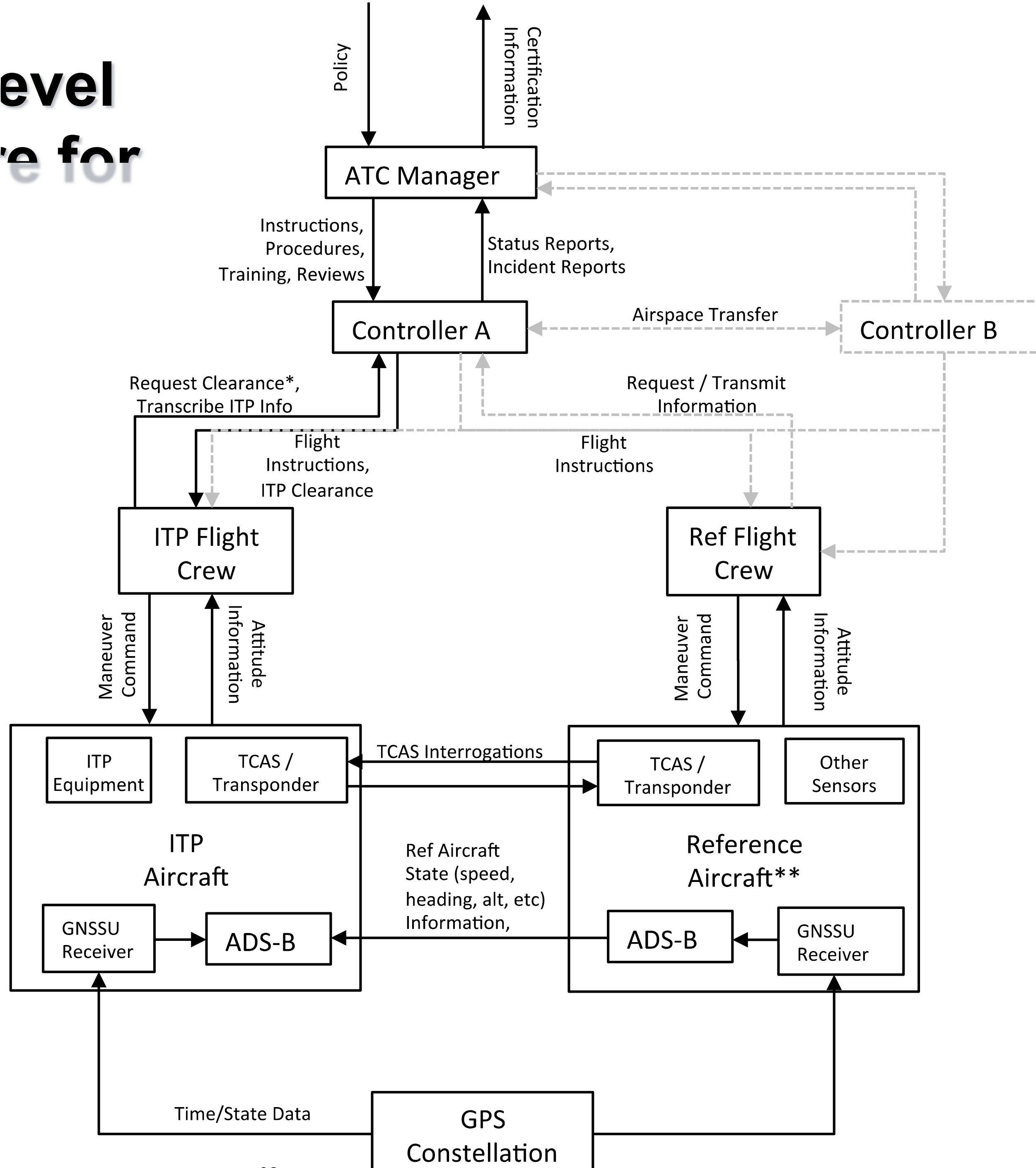
Control Structure Diagram – Level 0



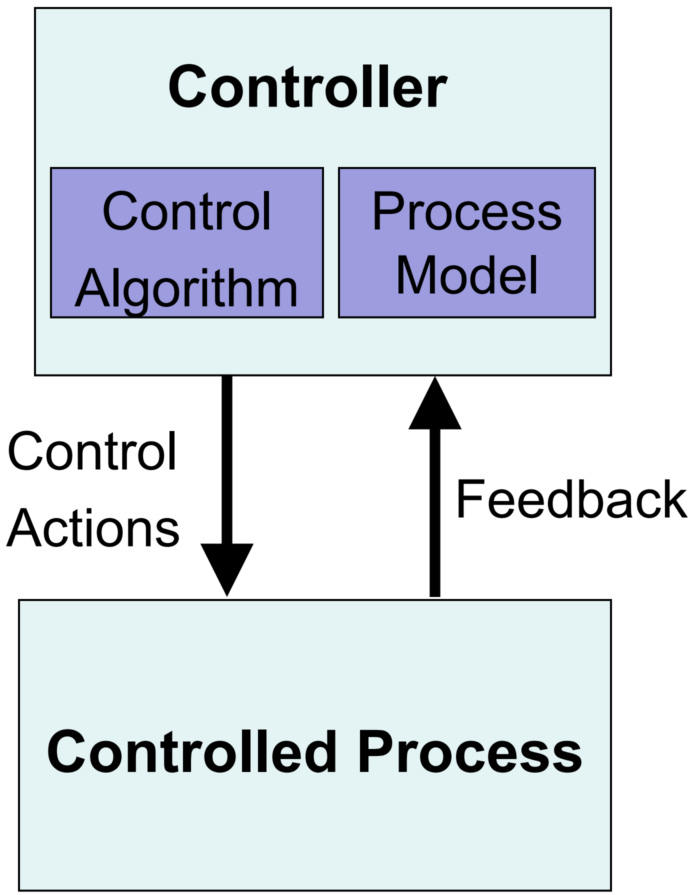
Control Structure Diagram – ISS Level 1



Example High-Level Control Structure for ITP



The Role of Process Models in Control



- Accidents often occur when process model inconsistent with state of controlled process (SA)
- A better model for role of software and humans in accidents than random failure model
- 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)

STAMP:
System-Theoretic Accident Model and Processes

Based on Systems Theory
(vs. Reliability Theory)

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 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
  - At Texas City, did not control the level of liquids in the ISOM tower;
  - In DWH, did not control the pressure in the well;
  - Financial system did not adequately control the use of financial instruments

Safety as a Dynamic Control Problem (2)

- Events are the result of the inadequate control
  - Result from lack of enforcement of safety constraints in system design and operations
- A change in emphasis:

"~~prevent~~ failures"

↓

"enforce safety constraints on system behavior"

Accident Causality Using STAMP