

Engineering a Safer World

Nancy Leveson
MIT



Presentation Outline

- Complexity in new systems reaching a new level (tipping point)
  - Old approaches becoming less effective
  - New causes of accidents not handled
- Need a paradigm change

  Change focus from

  Component reliability (reductionism)

  Systems thinking (holistic)

Presentation Outline

- STAMP: a new accident causality model based on systems theory (vs. reliability theory)
- More powerful tools based on STAMP
  - Hazard analysis
  - Accident/Incident Causal Analysis
  - Security
  - Others
- Does it work?
- Some current research topics

Why Our Efforts are Often Not Cost-Effective

- Efforts superficial, isolated, or misdirected
- Too much effort on assuring system safe vs. designing it to be safe
- Safety efforts start too late
- Inappropriate techniques for systems built today
- Focus efforts only on technical components (vs. human, management, organizational) and on system development (vs. operations)
- Systems assumed to be static through lifetime
- Limited learning from events

Why We Need a New Approach to Safety

"Without changing our patterns of thought, we will not be able to solve the problems we created with our current patterns of thought."

Albert Einstein

- Traditional safety engineering approaches developed for relatively simple electro-mechanical systems
- Accidents in complex, software-intensive systems are changing their nature
- Role of humans in systems is changing
- We need new ways to deal with safety in modern systems

The Starting Point:
Questioning Our Assumptions

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

(Attributed to many people)

Traditional Approach to Safety

- Traditionally view safety as a failure problem
  - Chain of directly related failure events leads to loss
- Forms the basis for most safety engineering and reliability engineering analysis:

  e,g, FTA, PRA, FMECA, Event Trees, etc.

  and design (establish barriers between events or try to prevent individual component failures:

  e.g., redundancy, overdesign, safety margins, interlocks, fail-safe design, …….

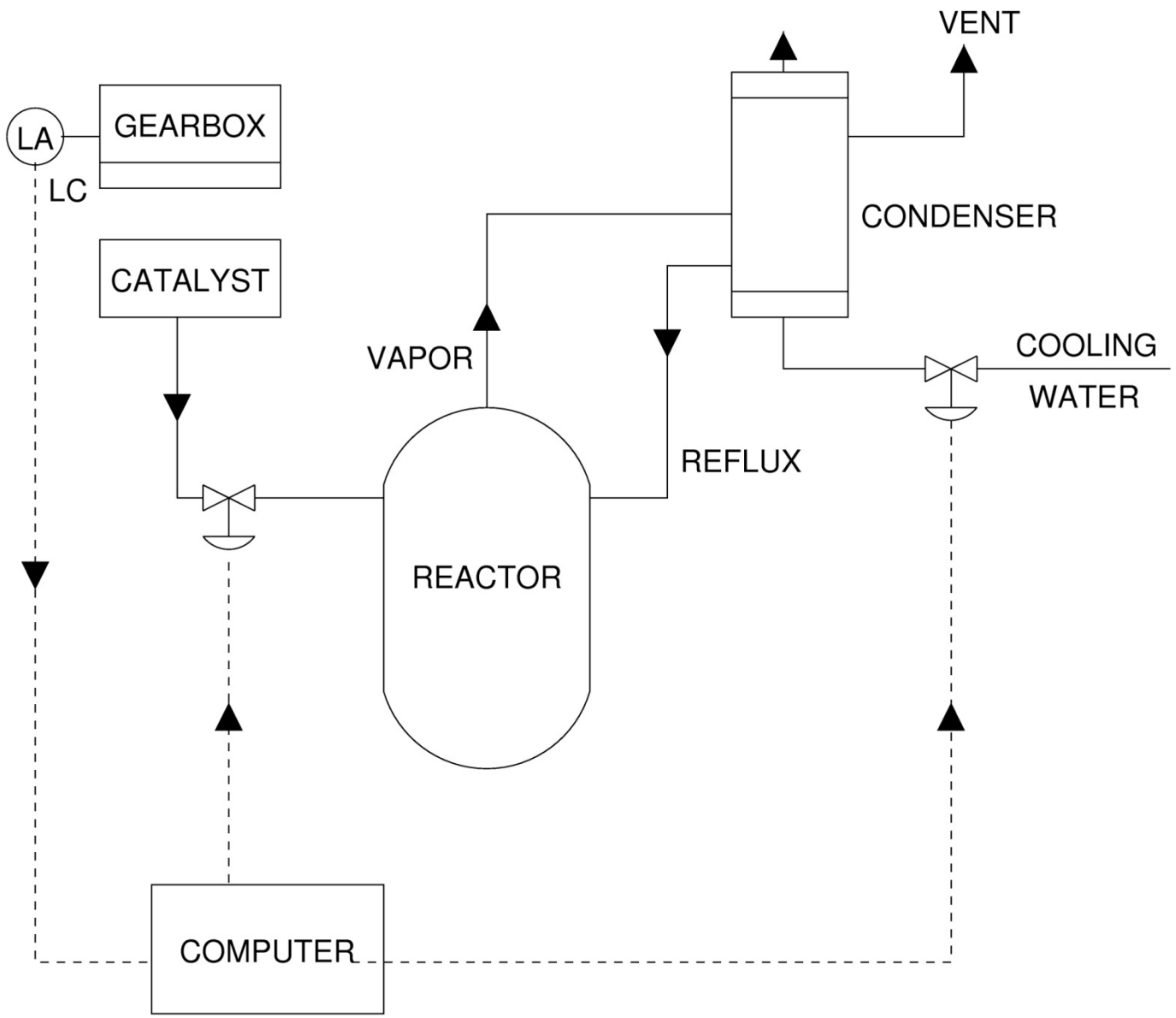
The Problem

- Chain-of-events model too simple for today's systems
  - Engineering has fundamentally changed in last 50 years
  - It is never going back
- Accident prevention/analysis techniques based on them will have limited usefulness
- We need something new

Safety ≠ Reliability

- Accidents happen with no component failures
- Components may fail with no accidents resulting

Accident with No Component Failures

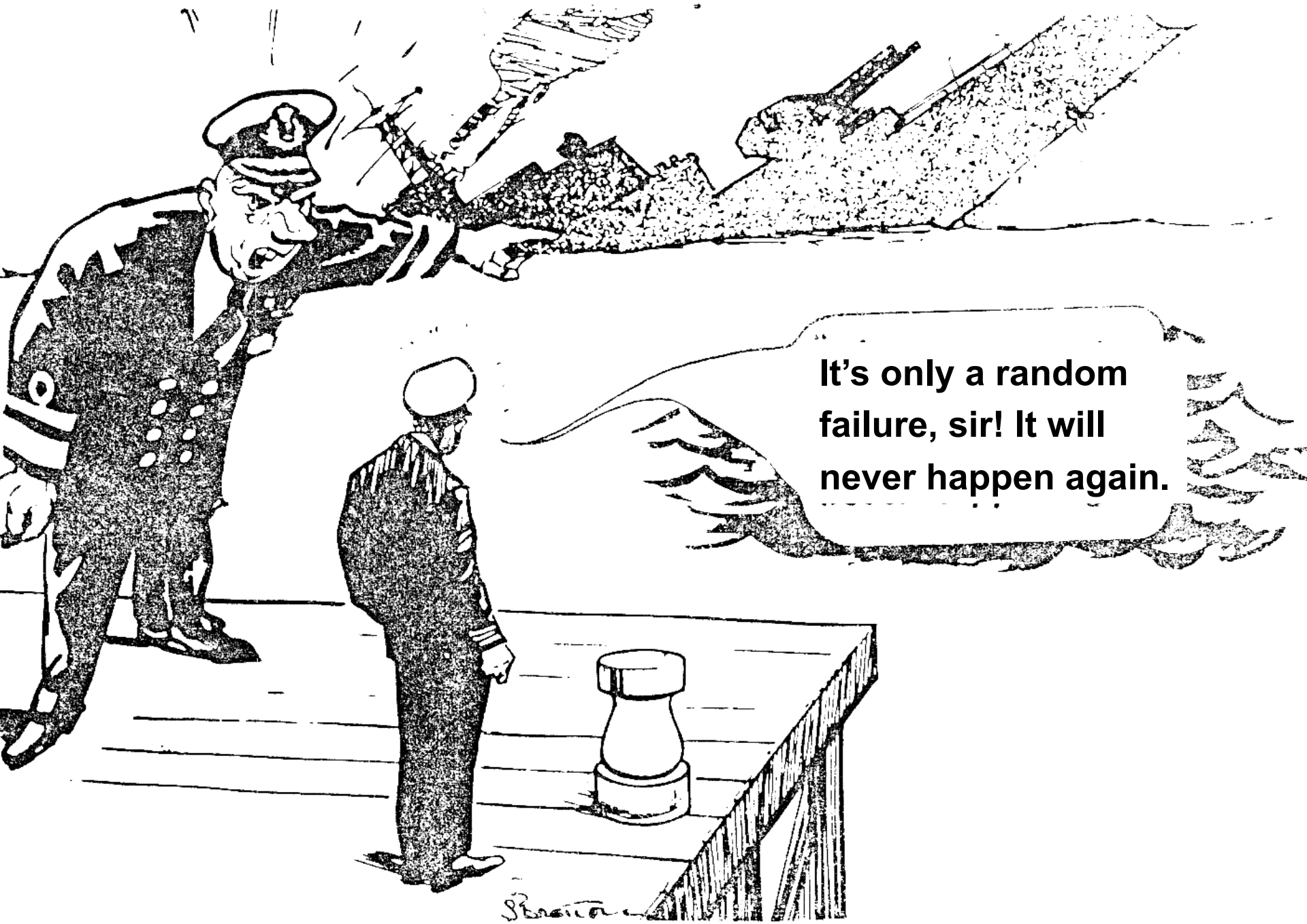


Types of Accidents

- Component Failure Accidents
  - Single or multiple component failures
  - Usually assume random failure
- Component Interaction Accidents
  - Arise in interactions among components
  - Related to interactive complexity and tight coupling
  - Exacerbated by introduction of computers and software but problem is system design errors

Relation of Complexity to Safety

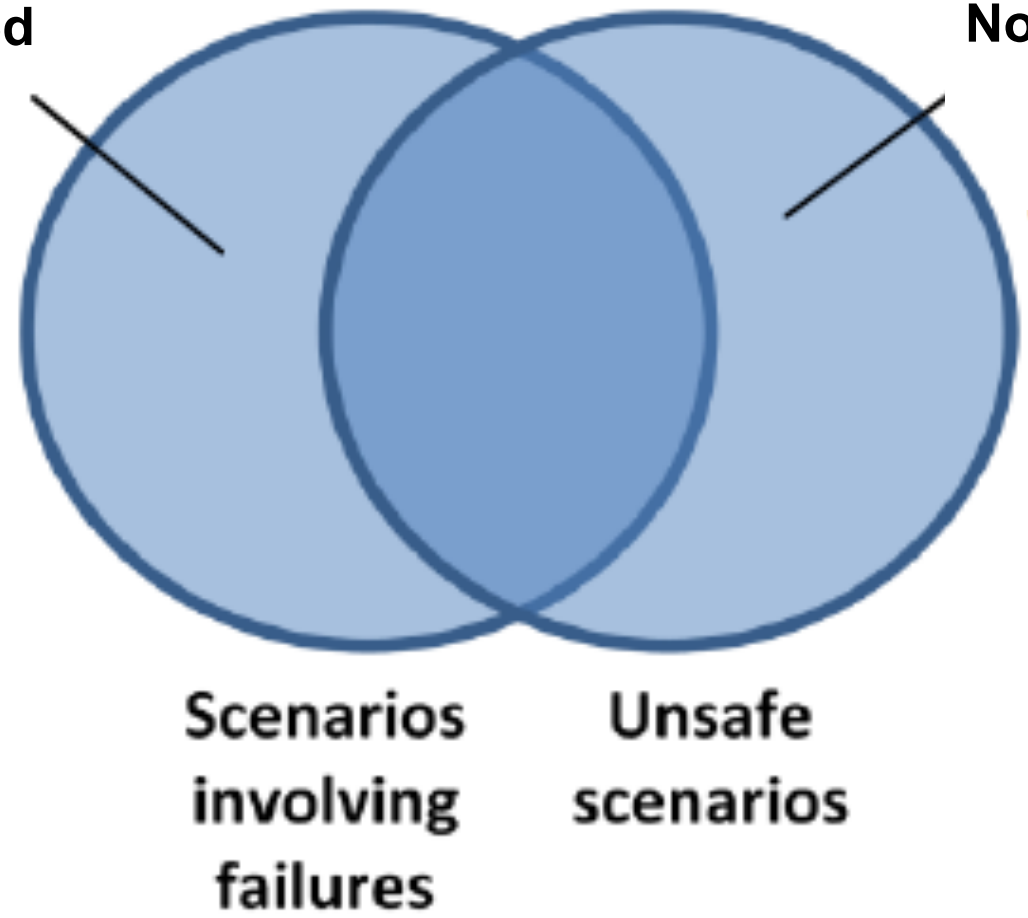
- In complex systems, behavior cannot be thoroughly
  - Planned
  - Understood
  - Anticipated
  - Guarded against
- Critical factor is intellectual manageability
- Leads to “unknowns” in system behavior
- Need tools to
  - Stretch our intellectual limits
  - Deal with new causes of accidents


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

Confusing Safety and Reliability

Not safety related

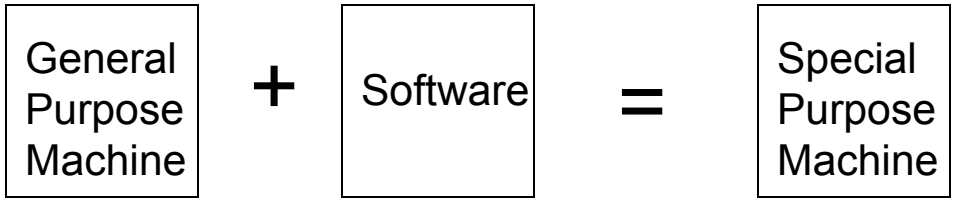
Not reliability related



Limitations of Chain-of-Events Causation Models

- Oversimplifies causality
- Excludes or does not handle
  - Component interaction accidents (vs. component failure accidents)
  - Indirect or non-linear interactions and complexity
  - Systemic factors in accidents
  - Human “errors”
  - System design errors (including software errors)
  - Adaptation and migration toward states of increasing risk

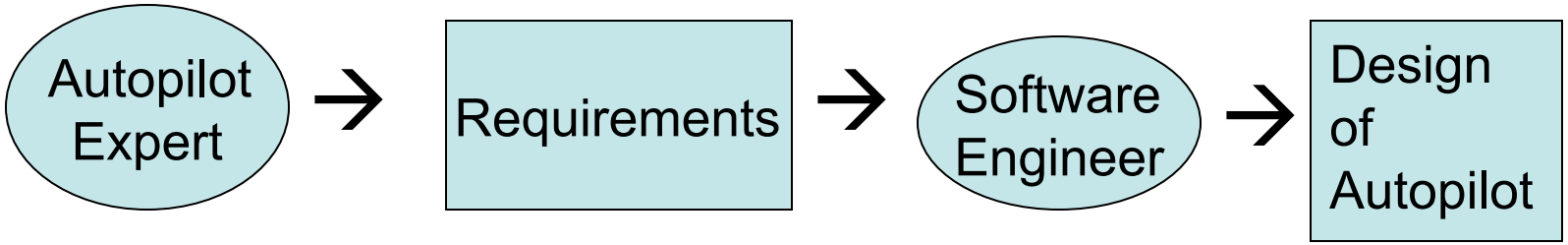
The Computer Revolution



- Software is simply the design of a machine abstracted from its physical realization
- Machines that were physically impossible or impractical to build become feasible
- Design can be changed without retooling or manufacturing
- Can concentrate on steps to be achieved without worrying about how steps will be realized physically

Abstraction from Physical Design

- Software engineers are doing physical design



- Most operational software errors related to requirements (particularly incompleteness)
- Software "failure modes" are different
  - Usually does exactly what you tell it to do
  - Problems occur from operation, not lack of operation
  - Usually doing exactly what software engineers wanted

Software-Related Accidents

- Are usually caused by flawed requirements
  - Incomplete or wrong assumptions about operation of controlled system or required operation of computer
  - Unhandled controlled-system states and environmental conditions

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

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 (admonish, fire, retrain them)
- Or do something about operators in general
  - Marginalize them by putting in more automation
  - Rigidify their work by creating more rules and procedures

Operator Error: Systems View

(Dekker, Rasmussen, Leveson, etc.)

- Operator error is a symptom, not a cause
- All behavior affected by context (system) in which occurs
- Role of operators in our systems is changing
  - Supervising rather than directly controlling
  - Systems are stretching limits of comprehensibility
  - Designing systems in which operator error inevitable and then blame accidents on operators rather than designers

Operator Error: Systems View (2)

- 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

What do we need to do?

- Expand our accident causation models
- Create new hazard analysis techniques
- Use new system design techniques
  - Safety-driven design
  - Integrate safety analysis and cognitive engineering into system engineering
- Improve accident analysis and learning from events
- Improve control of safety during operations
- Improve management decision-making and safety culture

STAMP: System-Theoretic Accident Model and Processes

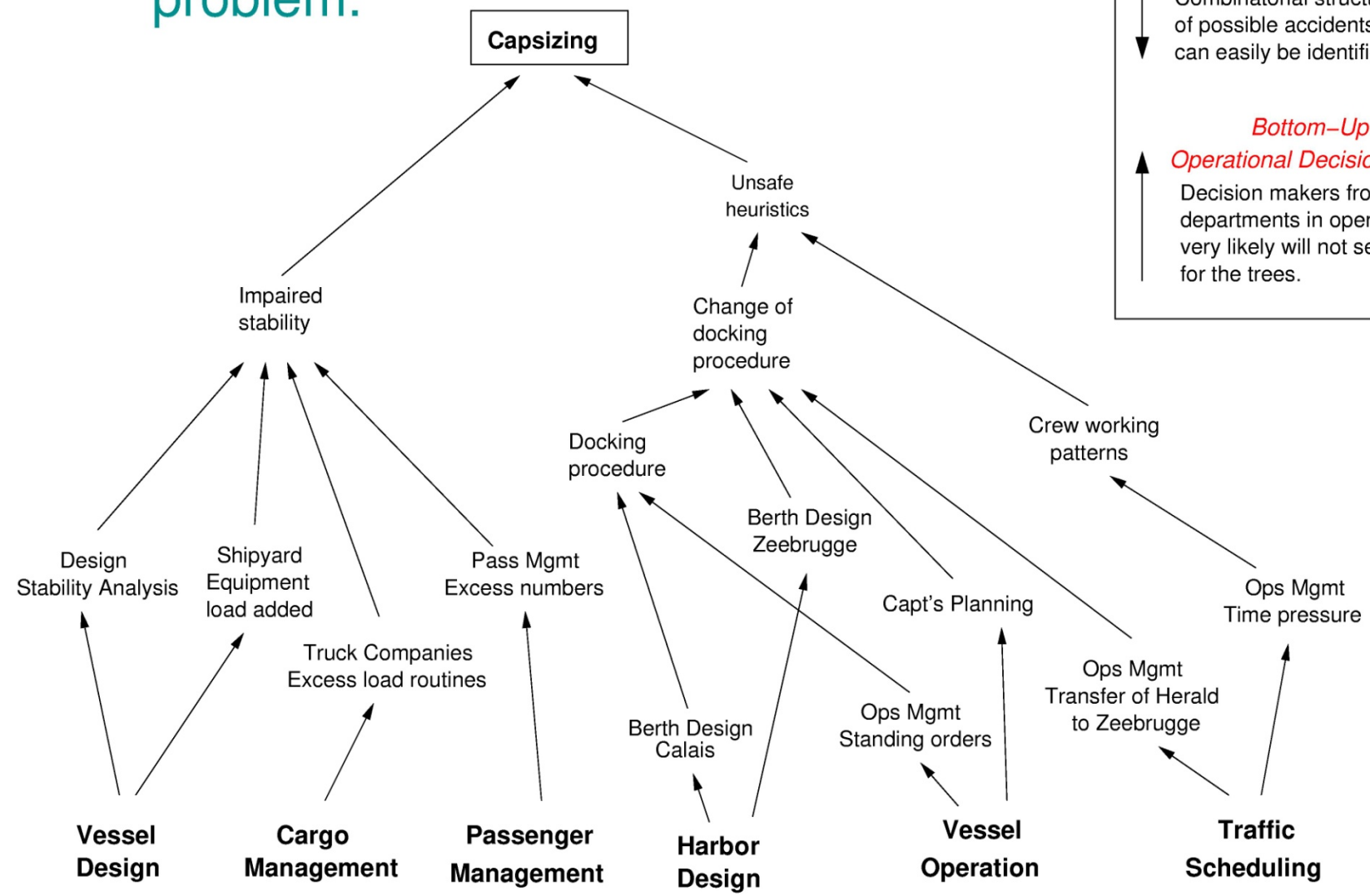
- Based on Systems Theory (not Reliability Theory)
- Applies systems thinking to safety

Safety and Security are System Properties

- Not in the individual components
- Arise when components (technical, physical, human) interact (emergent)
- Basing safety techniques on reliability theory limits the types of accidents and causes that can be handled

Safety is a system problem.

(From Rasmussen)



Top–Down Accident Analysis:
Combinatorial structure of possible accidents can easily be identified.

Bottom–Up Operational Decision Making:
Decision makers from separate departments in operational context very likely will not see the forest for the trees.

Reductionism vs. Systems Theory

- Three ways to deal with complexity
  - Analytic Reduction
  - Statistics
  - Systems Theory

Recommended reading: Peter Checkland, “Systems Thinking, Systems Practice,” John Wiley, 1981

Analytic Reduction

- Divide system into distinct parts for analysis

  Physical aspects → Separate physical components

  Behavior → Events over time

  Then examine parts separately

- Assumes such separation possible:

  1. The division into parts will not distort the phenomenon
     - Each component or subsystem operates independently
     - Analysis results not distorted when consider components separately

Analytic Reduction (2)

2. Components act the same when examined singly as when playing their part in the whole
   - or events not subject to feedback loops and non-linear interactions
3. Principles governing the assembling of components into the whole are themselves straightforward
   - Interactions among subsystems simple enough that can be considered separate from behavior of subsystems themselves
   - Precise nature of interactions is known
   - Interactions can be examined pairwise

Statistics

- Treat system as a structureless mass with interchangeable parts
- Use Law of Large Numbers to describe behavior in terms of averages
- Assumes components are sufficiently regular and random in their behavior that they can be studied statistically

Complex, Software-Intensive Systems

- 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

Systems Theory

- Developed for biology (von Bertalanffly) and engineering (Norbert Weiner) after World War II
- Basis of system engineering (ICBM systems of 1950's)
- Focuses on systems taken as a whole, not on parts taken separately
  - Some properties can only be treated adequately in their entirety, taking into account all social and technical aspects
  - These properties derive from relationships among the parts of the system

    How they interact and fit together
- A "top-down" approach to engineering (including safety and security)

STAMP Accident Causality Model

- Accidents (losses) 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

Examples of Safety Constraints

- Power must never be on when access door open
- Two aircraft must not violate minimum separation
- Aircraft must maintain sufficient lift
- Public health system must prevent exposure of public to contaminated water and food products
- Chemical plant (or nuclear plant) must prevent unintended release of toxins

STAMP (2)

- Losses involve a complex, dynamic “process”
  - Not simply chains of failure events
  - Arise in interactions among humans, machines and the environment
- Systems frequently migrate to states of higher risk
- A change in emphasis:

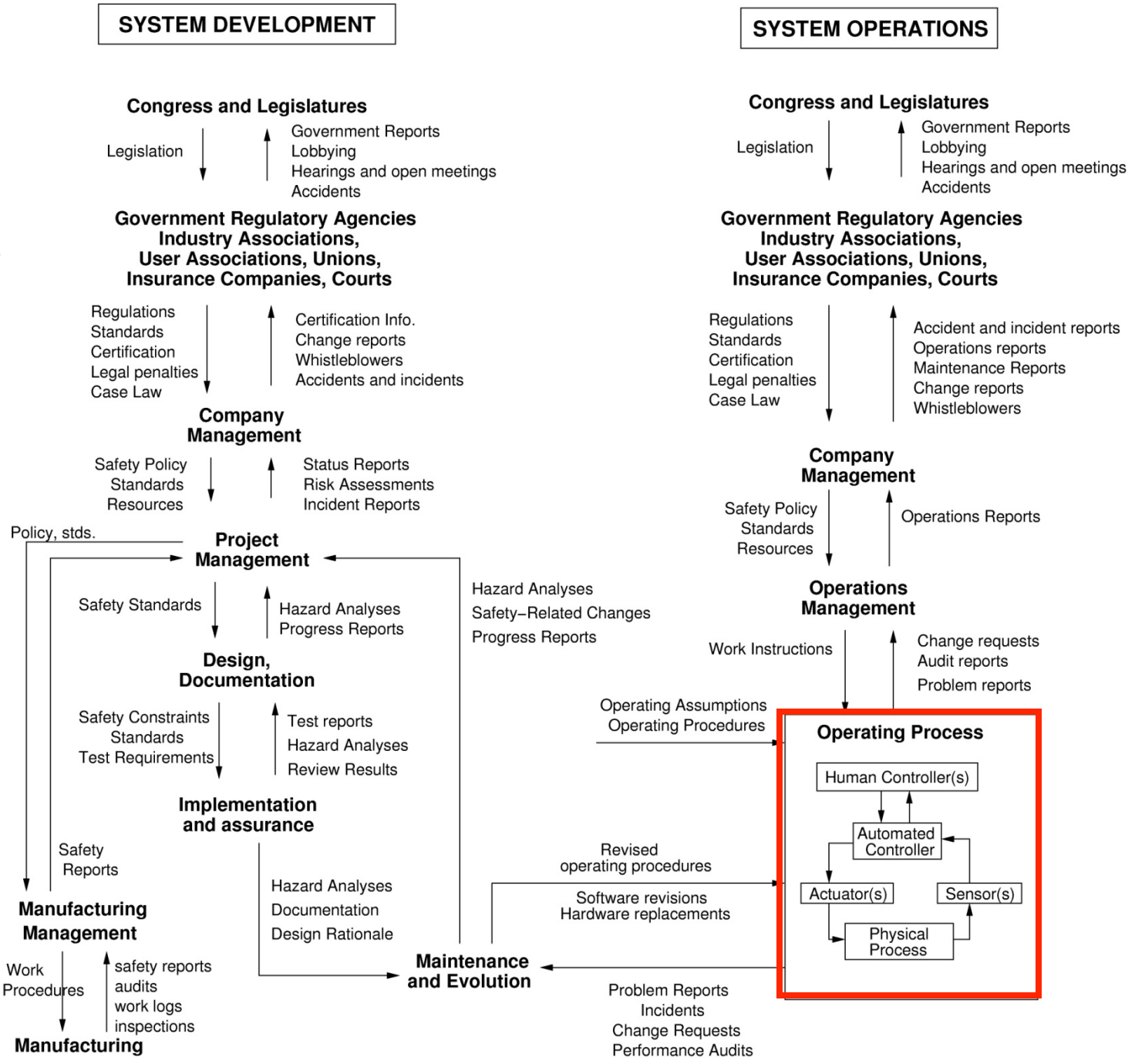
Safety as a Dynamic Control Problem

- 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 Fukushima, did not control the release of radioactivity from the plant
  - In DWH, did not control the pressure in the well
  - Financial system did not adequately control the use of financial instruments

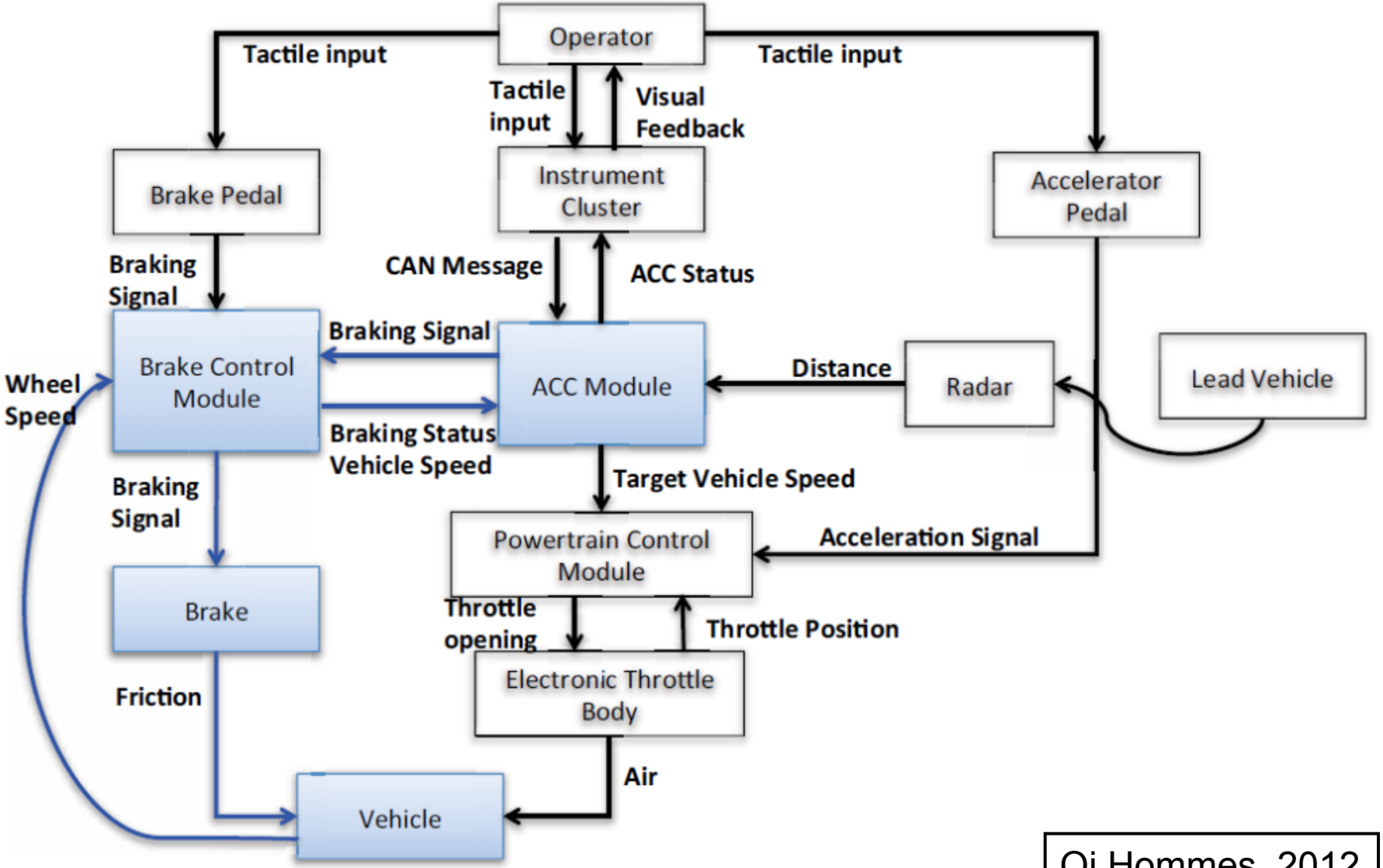
Safety as a Control Problem

- Identify the safety constraints
- Design a control structure to enforce constraints on system behavior and adaptation
  - Physical design (inherent safety)
  - Operations
  - Management
  - Social interactions and culture

Example Safety Control Structure

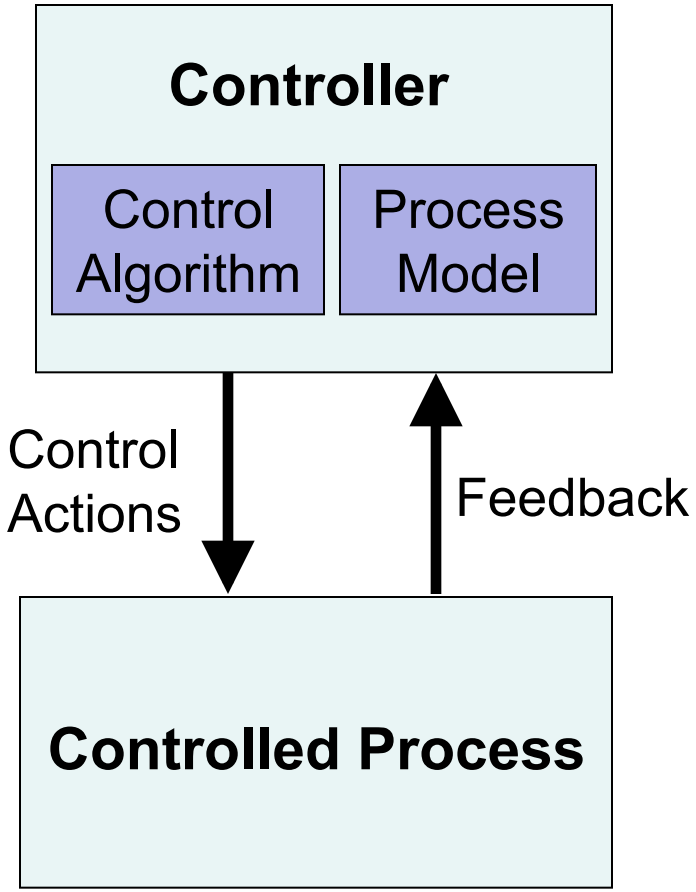


Example: ACC – BCM Control Loop



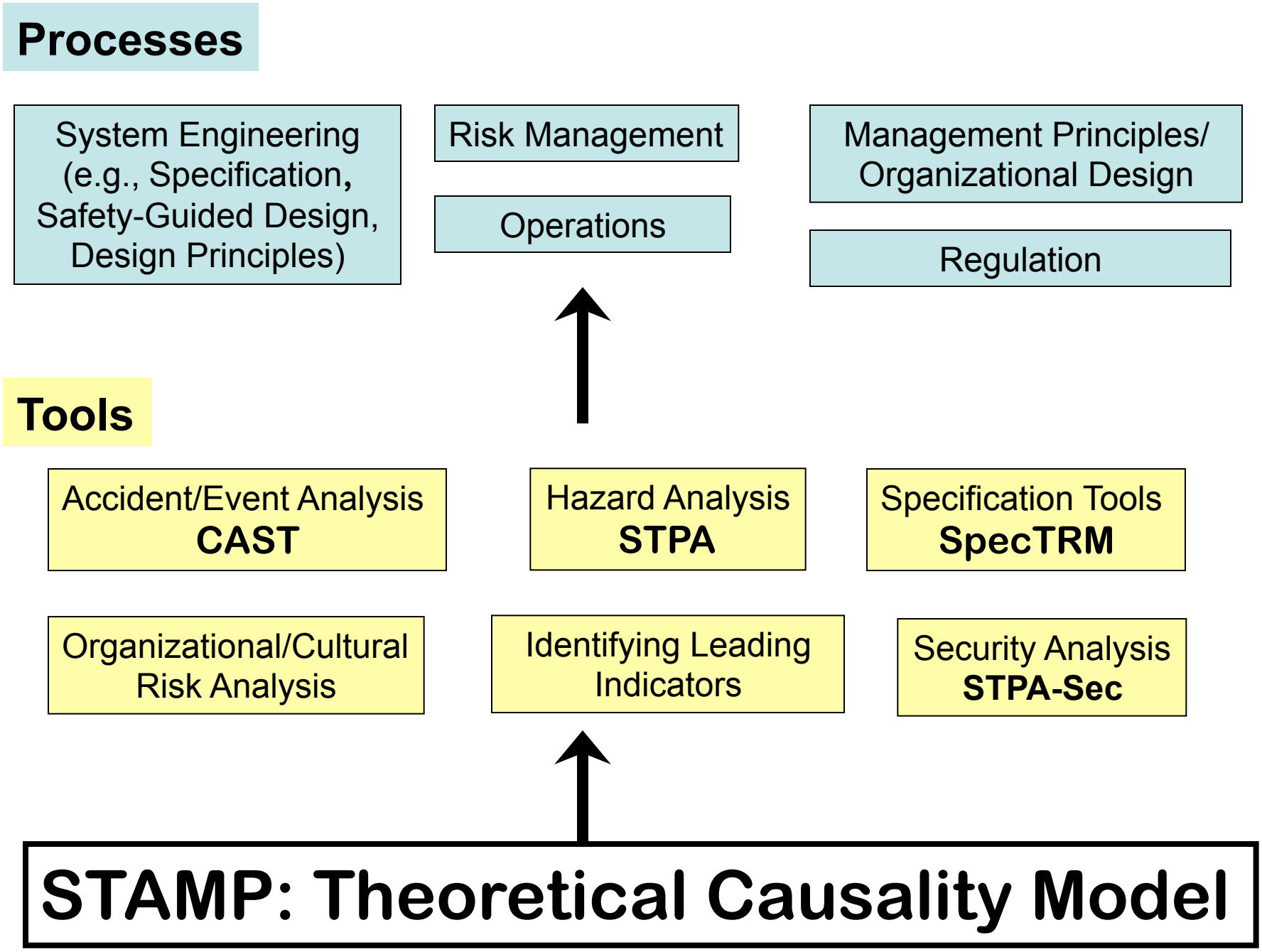
Qi Hommes, 2012

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)


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 STPA-Sec
STAMP: Theoretical Causality Model

STPA: System-Theoretic Process Analysis

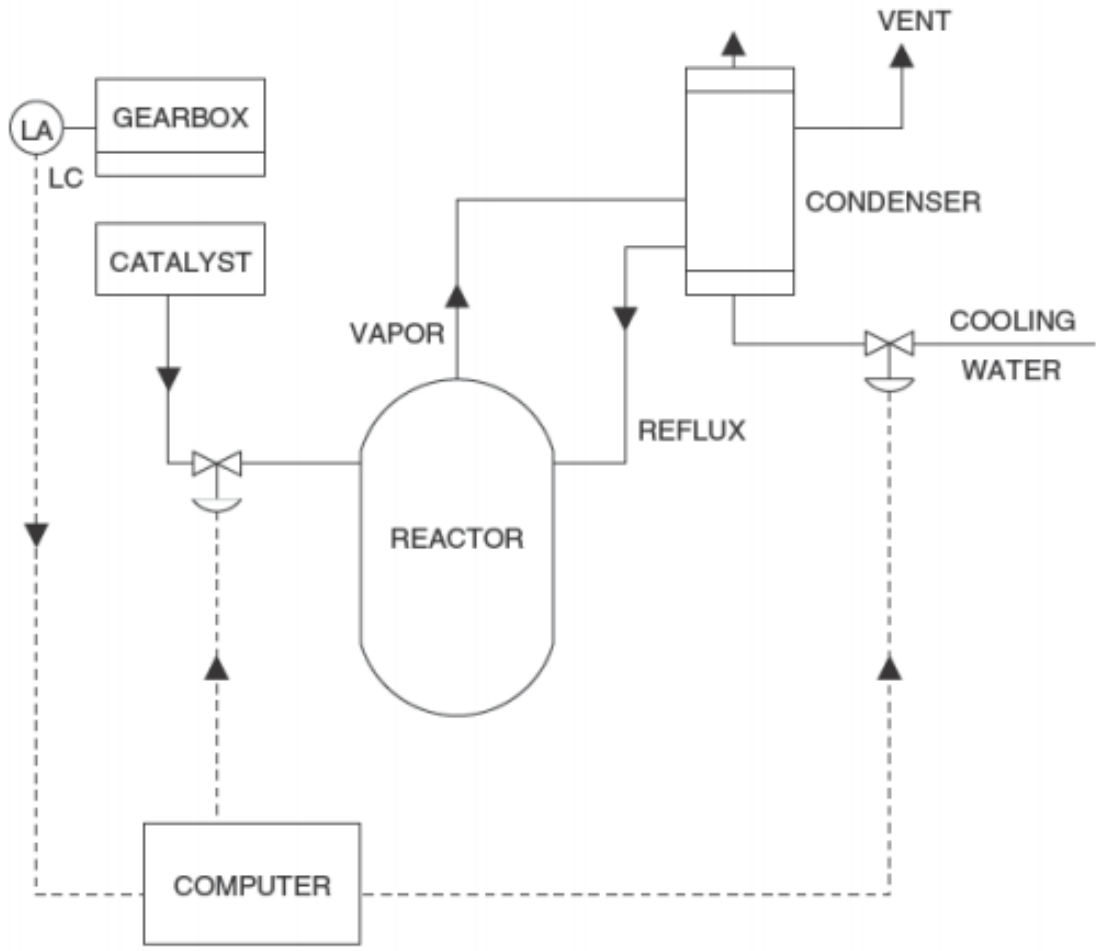
- Integrate safety and security into system engineering
  - Can be used from beginning of project
  - Safety-guided design:
    - Part of a top-down system engineering process
    - Start at very high-level of abstraction
    - Use STPA analysis to evaluate design decisions as they are being made
  - Guidance for evaluation and test
  - Can also be used for incident/accident analysis (to generate plausible scenarios)

STPA (2)

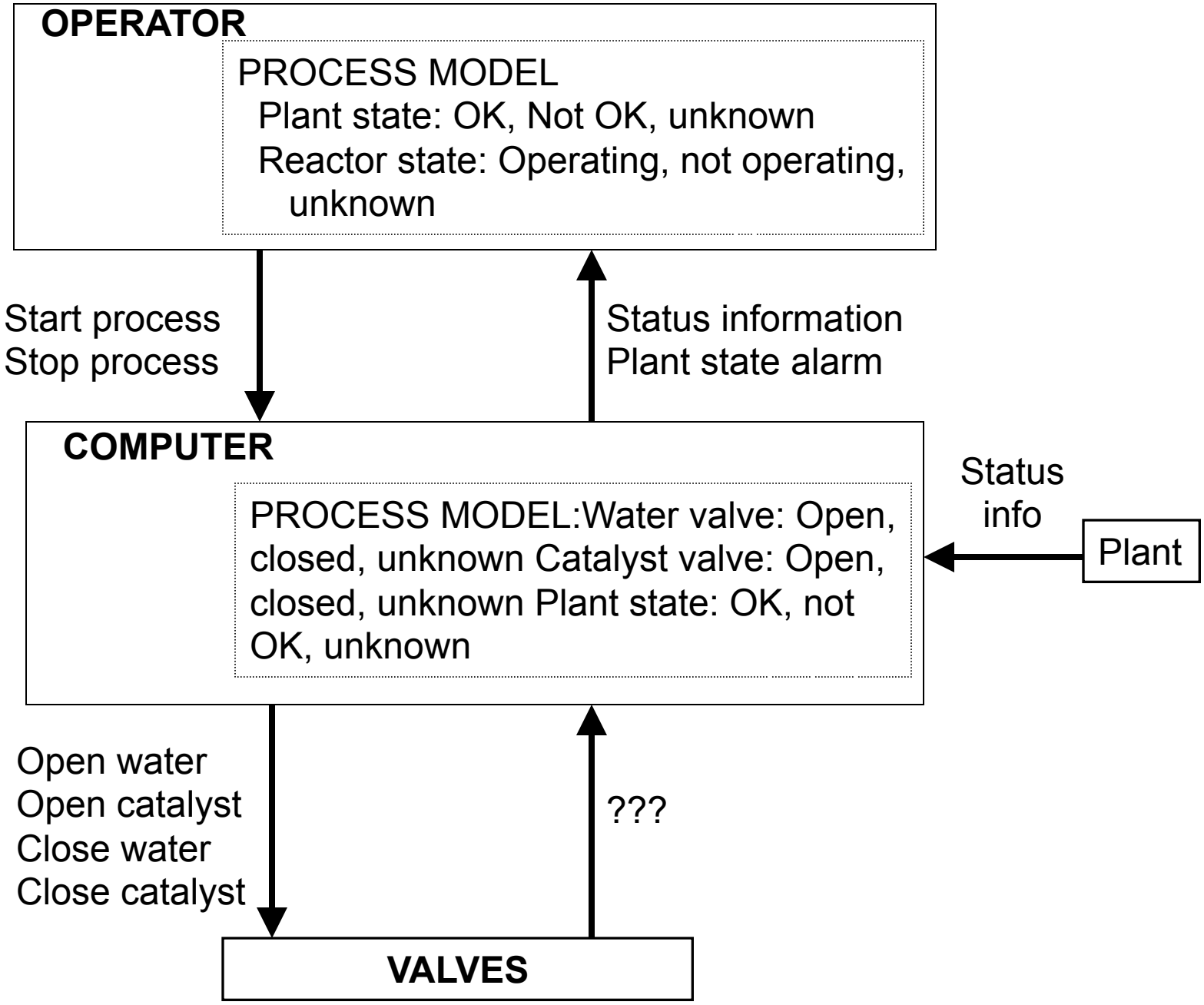
- Works also on social and organizational aspects of systems
- Generates system and component safety requirements (constraints)
- Identifies flaws in system design and scenarios leading to violation of a safety requirement (i.e., a hazard)

Steps in STPA

- Identify potential accidents/losses
- Identify hazards
- Construct functional control structure
- Identify unsafe control actions
- Generate system and component safety requirements
- Identify causal scenarios for unsafe control actions
- Augment system and component safety requirements and controls (mitigation) in system design



Create functional control structure for this physical structure


OPERATOR
PROCESS MODEL
Plant state: OK, Not OK, unknown
Reactor state: Operating, not operating, unknown
Start process
Stop process
Status information
Plant state alarm
COMPUTER
PROCESS MODEL:Water valve: Open, closed, unknown Catalyst valve: Open, closed, unknown Plant state: OK, not OK, unknown
Status info
Plant
Open water
Open catalyst
Close water
Close catalyst
???
VALVES

Identifying Unsafe Control Actions

Hazard: Catalyst in reactor without reflux condenser operating (water flowing through it)

	Not providing causes hazard	Providing causes hazard	Incorrect Timing/ Order	Stopped Too Soon / Applied too long
Open Water Valve	**Water valve not opened when catalyst open**	*[Conditions under which hazard results]*		
Close Water Valve				
Open Catalyst				

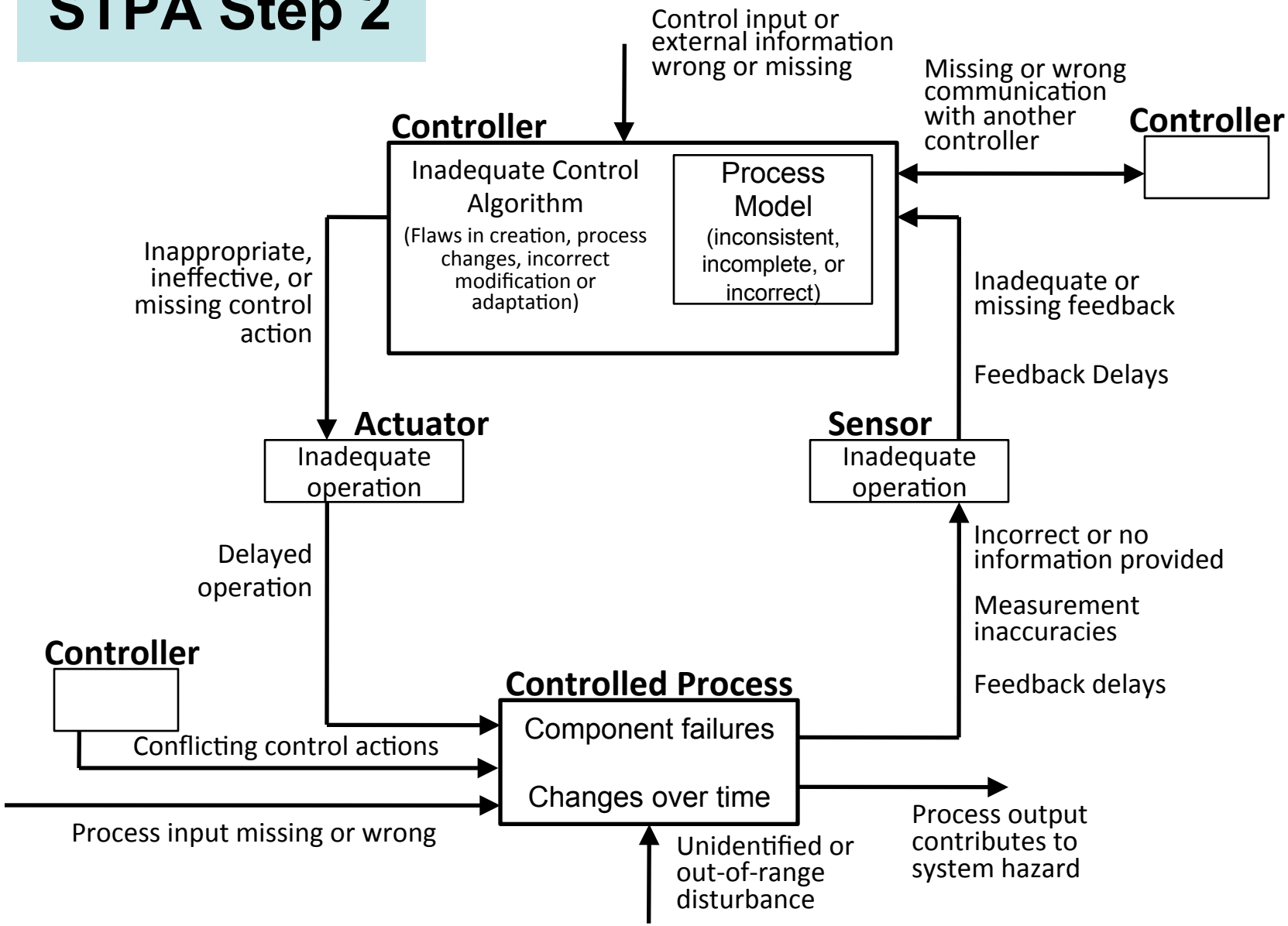
Hazard: Catalyst in reactor without reflux condenser operating (water flowing through it)

Control Action	Not providing causes hazard	Providing causes hazard	Too early/too late, wrong order	Stopped too soon/ applied too long
Open water	Not opened when catalyst open		Open water more than X seconds after open catalyst	Stop before fully opened
Close water		Close while catalyst open	Close water before catalyst closes	
Open catalyst		Open when water valve not open	Open catalyst more than X seconds before open water	
Close catalyst	Do not close when water closed		Close catalyst more than X seconds after close water	Stop before fully closed

What are the safety requirements (constraints) on the software controller given this table?

- Water valve must always be fully open before catalyst valve is opened.
  - Water valve must never be opened (complete opening) more than X seconds after catalyst valve opens
- Catalyst valve must always be fully closed before water valve is closed.
  - Catalyst valve must never be closed more than X seconds after water valve has fully closed.

STPA Step 2


Control input or external information wrong or missing
Controller
Inadequate Control Algorithm
(Flaws in creation, process changes, incorrect modification or adaptation)
Process Model
(inconsistent, incomplete, or incorrect)
Missing or wrong communication with another controller
Controller
Inappropriate, ineffective, or missing control action
Inadequate or missing feedback
Feedback Delays
Actuator
Inadequate operation
Sensor
Inadequate operation
Delayed operation
Incorrect or no information provided
Measurement inaccuracies
Feedback delays
Controller
Controlled Process
Component failures
Changes over time
Conflicting control actions
Process input missing or wrong
Unidentified or out-of-range disturbance
Process output contributes to system hazard

Exercise Continued (Batch Reactor)

- STEP 2: Identify some causes of the hazardous control action: *Open catalyst valve when water valve not open*
  - HINT: Consider how controller's process model could identify that water valve is open when it is not.
- What are some causes for a required control action (e.g., open water valve) being given by the software but not executed.
- What design features (controls) might you use to protect the system from the scenarios you found?

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 (GM, Ford, Nissan?)
  - Medical Devices and Hospital Safety
  - Chemical plants
  - Oil and Gas
  - Nuclear and Electrical Power
  - $C0_2$ Capture, Transport, and Storage
  - Etc.

Is it Practical? (2)

Social and Managerial

- Analysis of the management structure of the space shuttle program (post-Columbia)
- Risk management in the development of NASA's new manned space program (Constellation)
- NASA Mission control ─ re-planning and changing mission control procedures safely
- Food safety
- Safety in pharmaceutical drug development
- Risk analysis of outpatient GI surgery at Beth Israel Deaconess Hospital
- Analysis and prevention of corporate fraud

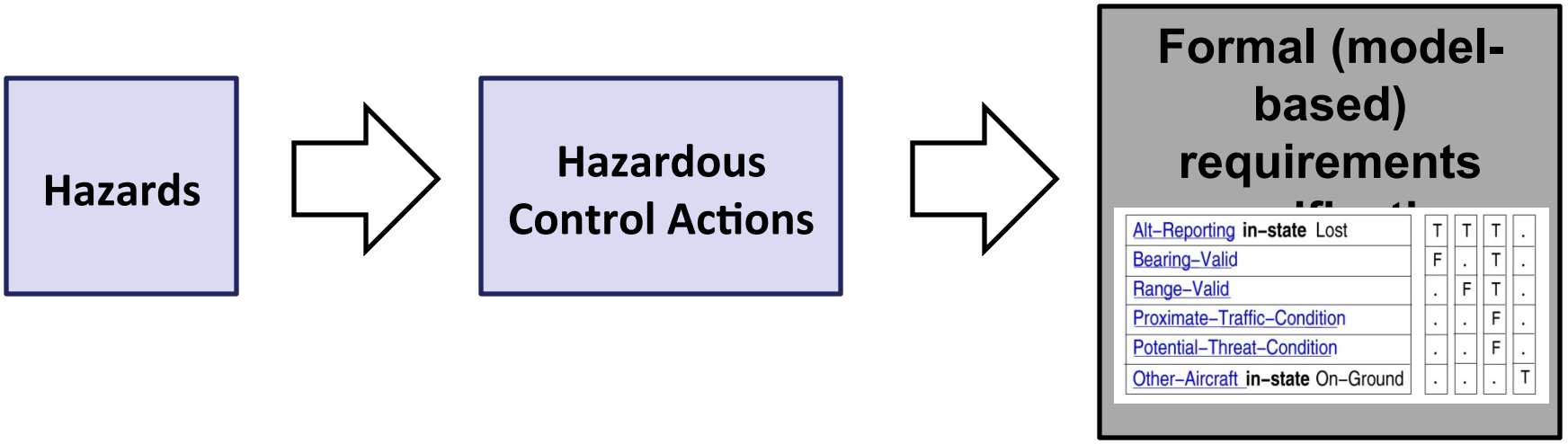
Does it Work?

- Most of these systems are very complex (e.g., the U.S. Missile Defense System)
- In all cases where a comparison was made:
  - STPA found the same hazard causes as the old methods
  - Plus it found more causes than traditional methods
    - All components were operating exactly as intended but complexity of component interactions led to unanticipated system behavior
    - Examples: missing case in software requirements, timing problems in sending and receiving messages, etc.
  - Sometimes found accidents that had occurred that other methods missed
  - Cost was orders of magnitude less than the traditional hazard analysis methods

One Example:

- Blood Gas Analyzer (Vincent Balgos)
  - 75 scenarios found by FMEA
  - 175 identified by STPA
  - Took much less time and resources (mostly human)
    - FMEA took a team of people months to perform
    - STPA took one person two weeks (and he was just learning STPA)
  - Only STPA found scenario that had led to a Class 1 recall by FDA (actually found nine scenarios leading to it)

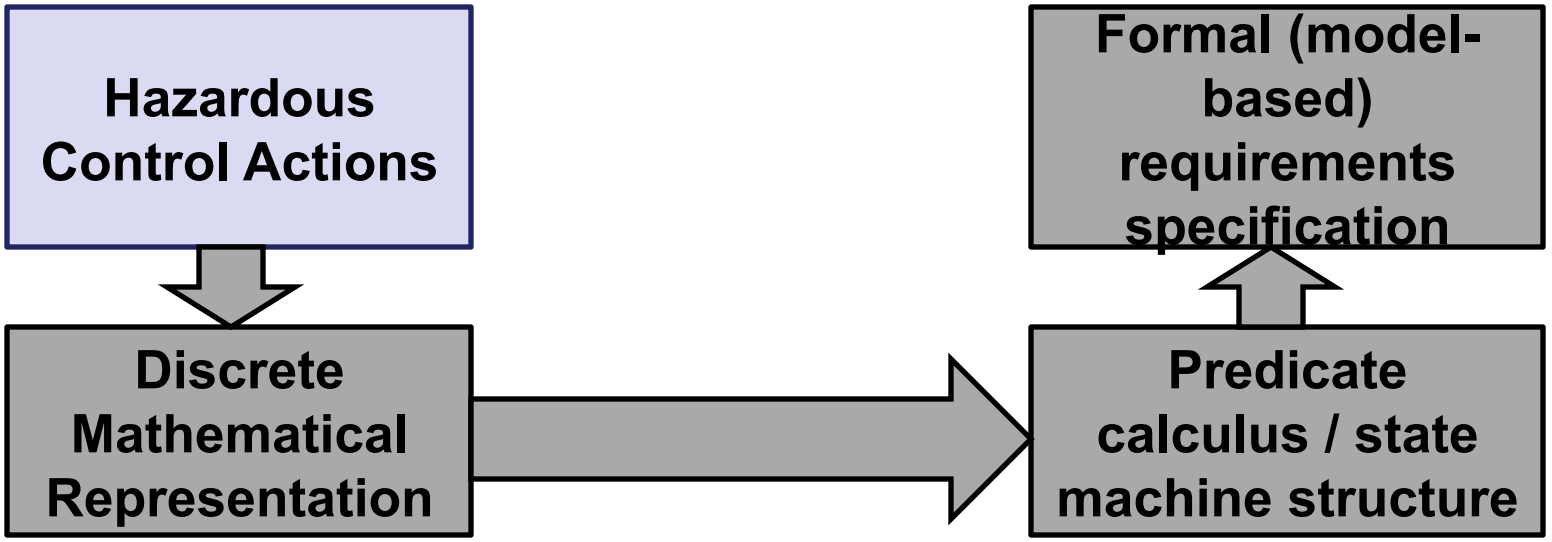
Automating STPA (John Thomas)



- Can automate most of Step 1 (but requires human decision making)
- Formal underlying discrete mathematical models allow for automated consistency/completeness checks (can detect conflicts)
- Have not yet automated Step 2 (causes of unsafe control actions)

Generating safety requirements

- Formal requirements can be derived using
  - Discrete mathematical structure for hazardous control actions
  - Predicate calculus to obtain necessary requirements
- Automatically generate formal requirements given these relationships!



STPA Primer

- Examples, exercises
- http://sunnyday.mit.edu/STPA-Primer-v0.pdf
- More to come

CAST (Causal Analysis using STAMP)

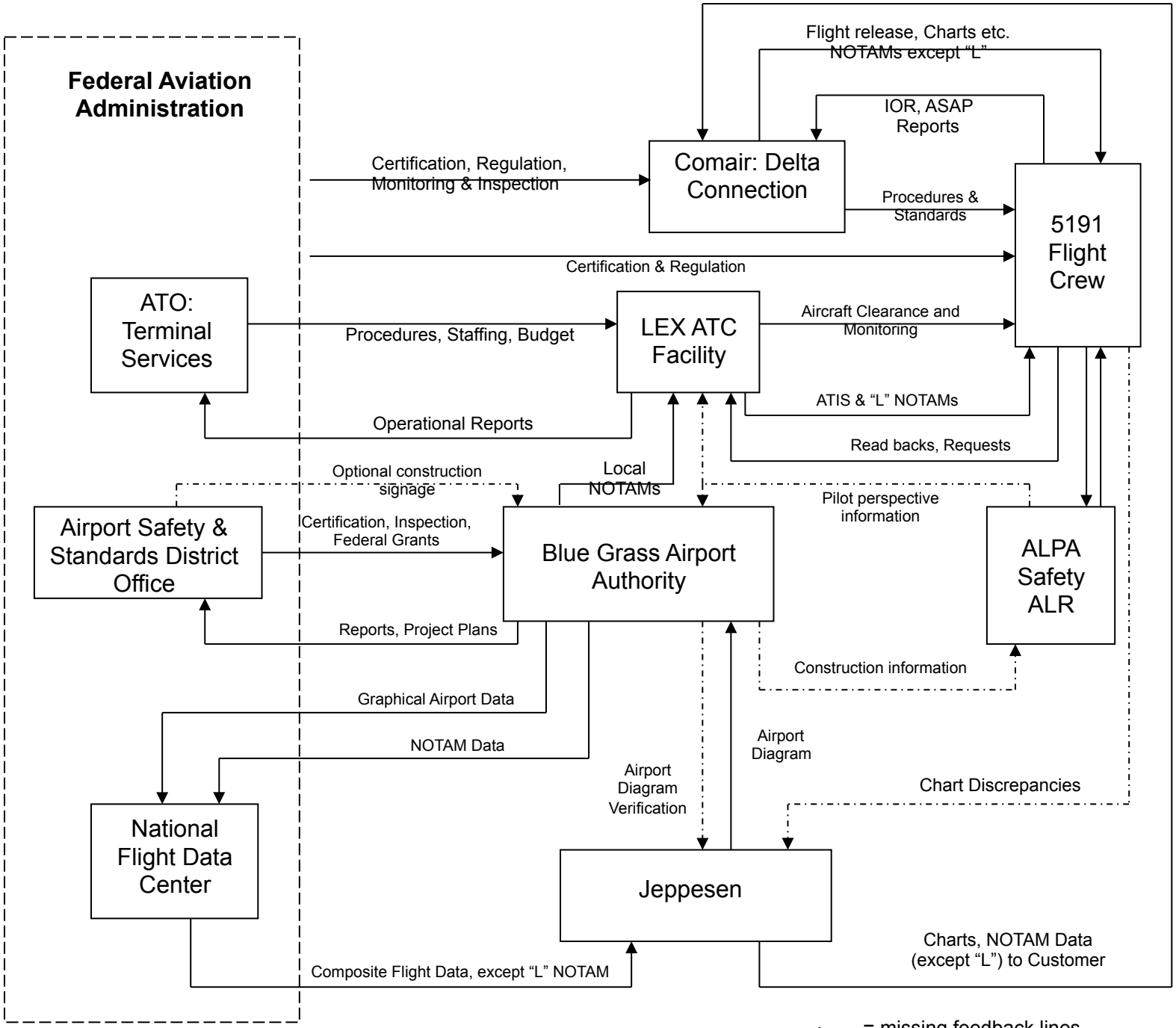
- A “why” analysis, not a “blame” analysis
- Identify system hazard violated and the system safety design constraints
- Construct the safety control structure as it was designed to work
  - Component responsibilities (requirements)
  - Control actions and feedback loops
- For each component, determine if it fulfilled its responsibilities or provided inadequate control.
  - If inadequate control, why? (including changes over time)
    - Context
    - Process Model Flaws
- For humans, why did it make sense for them to do what they did (to reduce hindsight bias)

CAST (2)

- Examine coordination and communication
- Consider dynamics and migration to higher risk
- Determine the changes that could eliminate the inadequate control (lack of enforcement of system safety constraints) in the future.
- Generate recommendations
- Continuous Improvement
  - Assigning responsibility for implementing recommendations
  - Follow-up to ensure implemented
  - Feedback channels to determine whether changes effective
    - If not, why not?

ComAir 5191 (Lexington) Sept. 2006

Analysis using CAST by Paul Nelson,
ComAir pilot and human factors expert
(for report: http://sunnyday.mit.edu/papers/nelson-thesis.pdf


Federal Aviation Administration
Flight release, Charts etc.
NOTAMs except "L"
IOR, ASAP Reports
Certification, Regulation, Monitoring & Inspection
Comair: Delta Connection
Procedures & Standards
5191 Flight Crew
Certification & Regulation
ATO: Terminal Services
Procedures, Staffing, Budget
LEX ATC Facility
Aircraft Clearance and Monitoring
ATIS & "L" NOTAMs
Operational Reports
Read backs, Requests
Optional construction signage
Local NOTAMs
Pilot perspective information
Airport Safety & Standards District Office
Certification, Inspection, Federal Grants
Blue Grass Airport Authority
ALPA Safety ALR
Reports, Project Plans
Construction information
Graphical Airport Data
NOTAM Data
Airport Diagram
Airport Diagram Verification
Chart Discrepancies
National Flight Data Center
Jeppesen
Charts, NOTAM Data (except "L") to Customer
Composite Flight Data, except "L" NOTAM
= missing feedback lines

Evaluating CAST on Real Accidents

- Used on many types of accidents
  - Aviation
  - Trains (Chinese high-speed train accident)
  - Chemical plants and off-shore oil drilling
  - Road Tunnels
  - Medical devices
  - Etc.
- All CAST analyses so far have identified important causal factors omitted from official accident reports

Evaluations (2)

- Jon Hickey, US Coast Guard applied to aviation training accidents
  - US Coast Guard currently uses HFACS (based on Swiss Cheese Model)
  - Spate of recent accidents but couldn't find any common factors
  - Using CAST, found common systemic factors not identified by HFACS
  - USCG now deciding whether to adopt CAST

Integrated Approach to Safety and Security:

- 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
  - An integrated approach to safety and security is possible
  - New paradigm for safety will work for security too

Top-Down Approach

- Starts with identifying losses
- Identify vulnerabilities and system safety/security constraints
- Build functional control model
  - Controlling constraints whether safety or security
  - Includes physical, social, logical and information, operations, and management aspects
- Identify unsafe/unsecure control actions and causes for them
  - May have to add new causes, but rest of process is the same

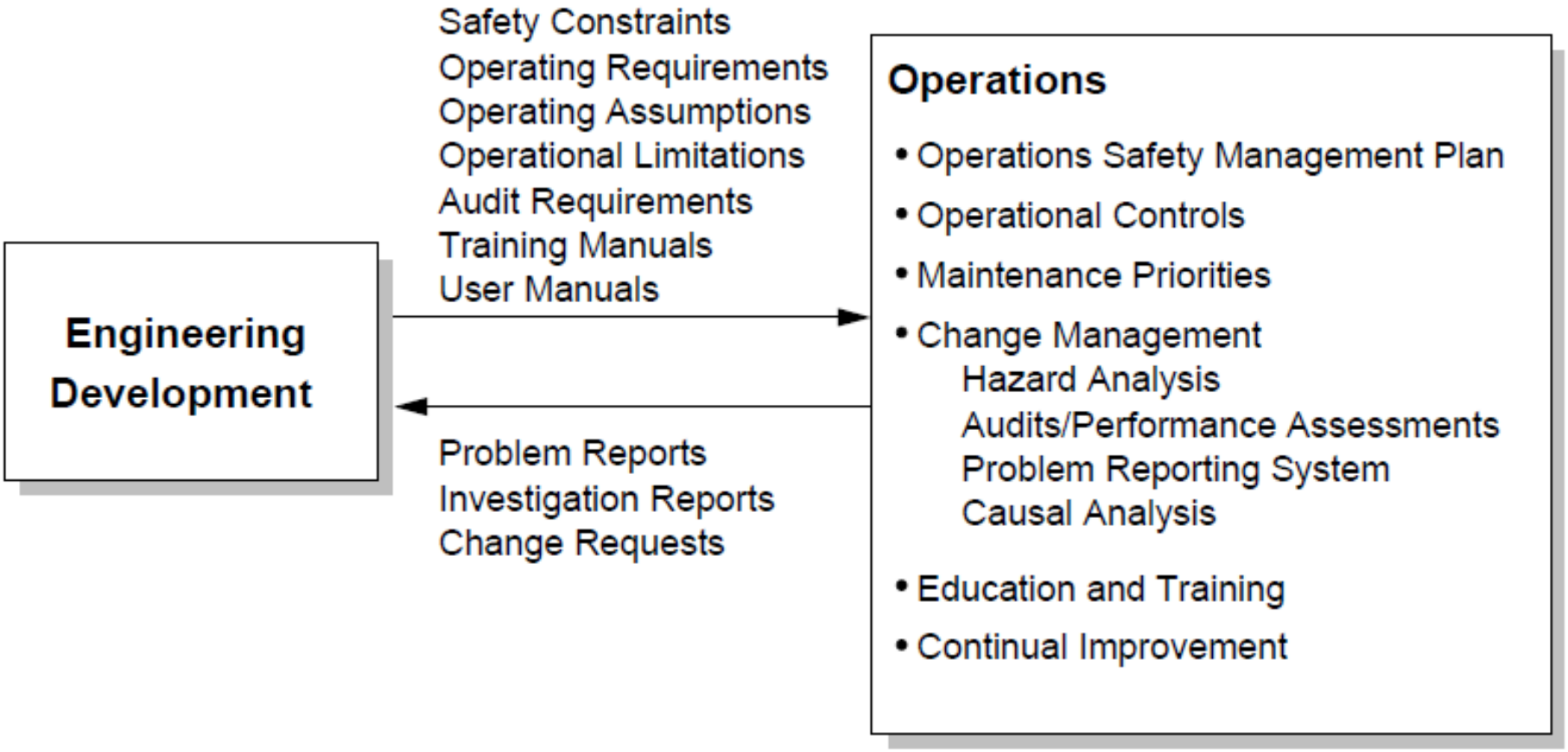
Example: Stuxnet

- 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 cause:
  - *Incorrect process model*: thinks spinning at less than maximum speed
  - Could be inadvertent or advertent

Evaluation of STPA-Sec

- Informal so far but with real red teams
  - Went through STPA-Sec steps
  - Found things they had not thought of before
- Formal experiment in Spring 2014

Safety in Operations



Safety Management and Safety Culture

- Why managers should care about safety
- How to achieve project and company safety goals
- Designing an effective safety control structure

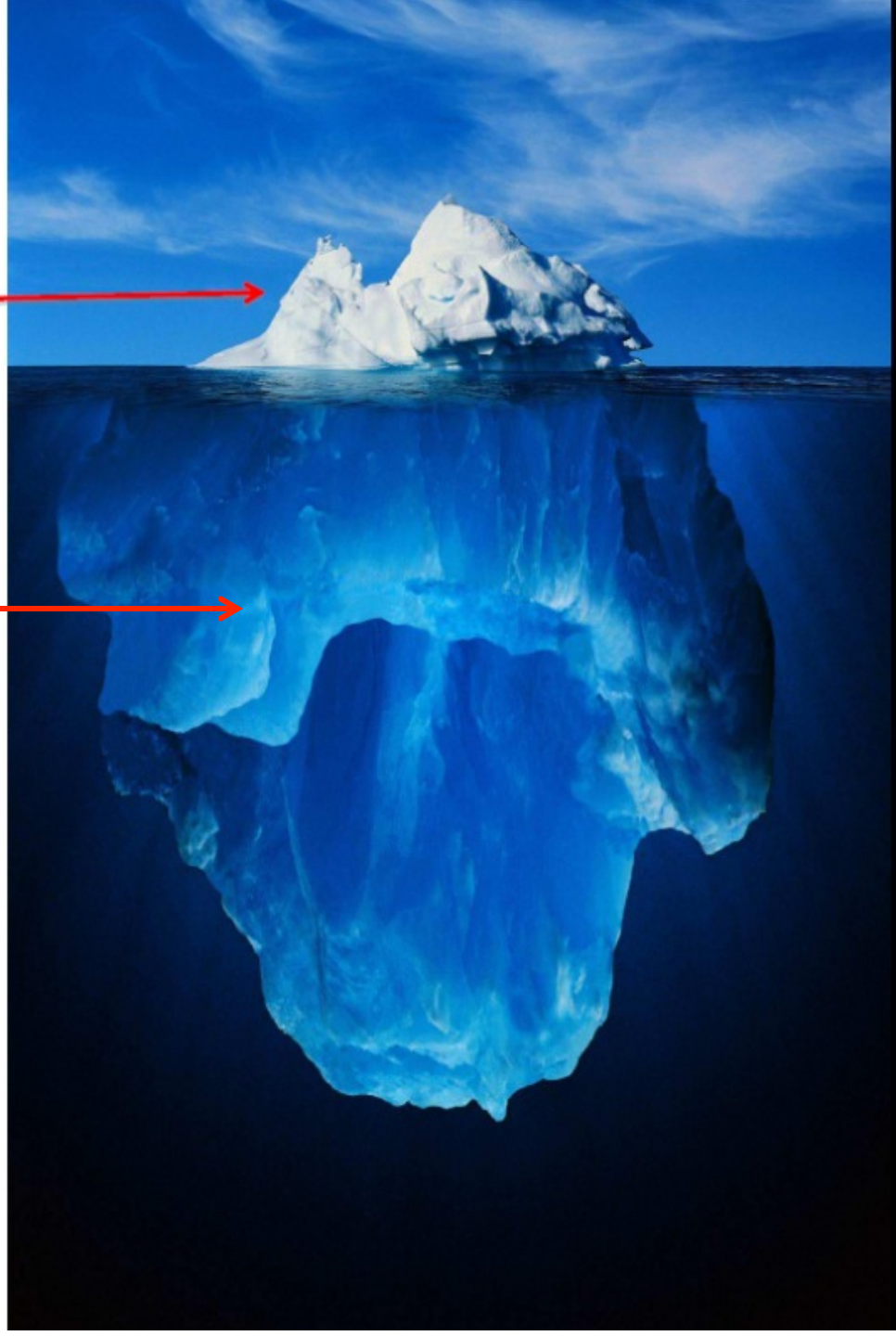
Summary

- More comprehensive and powerful approach to safety (and security)
  - Examines inter-relationships rather than just linear cause-effect chains.
  - Includes what consider now (component failures) but more (e.g., system design errors, requirements flaws)
- Includes social, human, software-related factors
- Top-down system engineering approach
  - Safety-guided design starts early at concept formation
  - Generates safety requirements from hazard analysis
- Handles much more complex systems than traditional safety analysis approaches

LEVERAGE

Event-based thinking

Systems Thinking



WYOUNG@MIT.EDU

Current Research Projects

- Applications: NextGen (ATM), UAVs, Railroads, Healthcare, Autos, …
- STPA-SDD (Safety-Driven Design) and Model-Based System Conceptual Development
- Safety analysis of radiation therapy procedures at U.C. San Diego Medical Center
- Hospital ICU Safety and Adverse Event Causal Analysis
- Analyzing Feature Interaction in Automobiles
- Integration of UAVs (RPVs) into the NAS (National Airspace System)
- Adding more sophisticated human factors analysis to STPA
- Risk management and managerial decision making (visualization of risk)
- Security (cyber and physical)
- Automated Tools

Tutorials

- STPA (Hazard Analysis): John Thomas, 54-100
- CAST (Accident/Incident Analysis): Paul Nelson, 56-154
- Security: Bill Young, Adam Williams, Michael Stone (Akamai)
- Experienced Users Meeting