Massachusetts
Institute of
Technology

2024 STAMP Workshop

STPA Introduction

Dr. John Thomas

Any questions? Email me! JThomas4@mit.edu

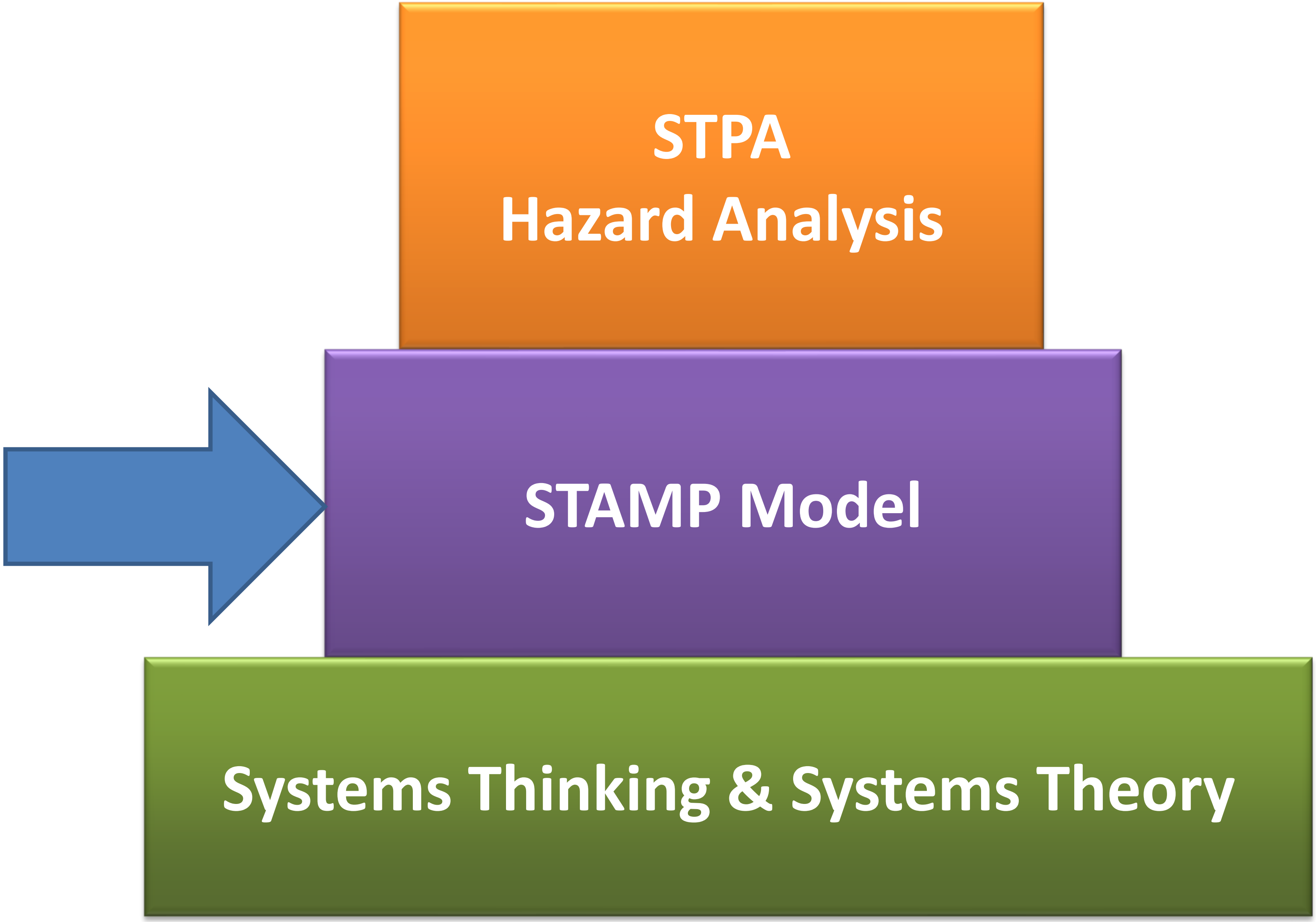
Tutorial Objective

- These short tutorials are **<u>not training classes</u>**
- We cannot cover everything in these tutorial sessions.
- The objective is to introduce some of the core concepts so new attendees can follow the presentations this week.
- Training and practice with a qualified instructor are needed to apply these techniques and become proficient (as with most techniques). These short tutorials are subsets of larger training classes.

This STPA introduction is **not a training class**

	Today	Actual Training & Collaborative Projects
Goal	Provide basic STPA familiarity to follow the presentations	Build capability to apply STPA proficiently to a real system.
Duration	< 5.5 hours	Training: ~40 hours of hands-on instruction
Hands-on Practice	Minimal	Extensive, using real-world applications
Complexity of Examples	Minimal	Moderate - High
Analysis Depth/Quality	Superficial	High-quality, correct, and careful analysis. Details matter. Will generate new engineering insights, uncover new flaws, produce real technical requirements.
Exit Criteria	Clock = 10:30	Participants demonstrate proficiency applying STPA themselves on a real system, satisfy 25 certification criteria, and receive a certificate
Instructor Feedback Loop	Minimal	Loop: - Introduce new step / concept - Practice new step / concept - Performance reviewed - Gaps in skill and knowledge identified - Corrections made - Repeat

System Theory, STAMP, STPA



STPA
Hazard Analysis

STAMP Model

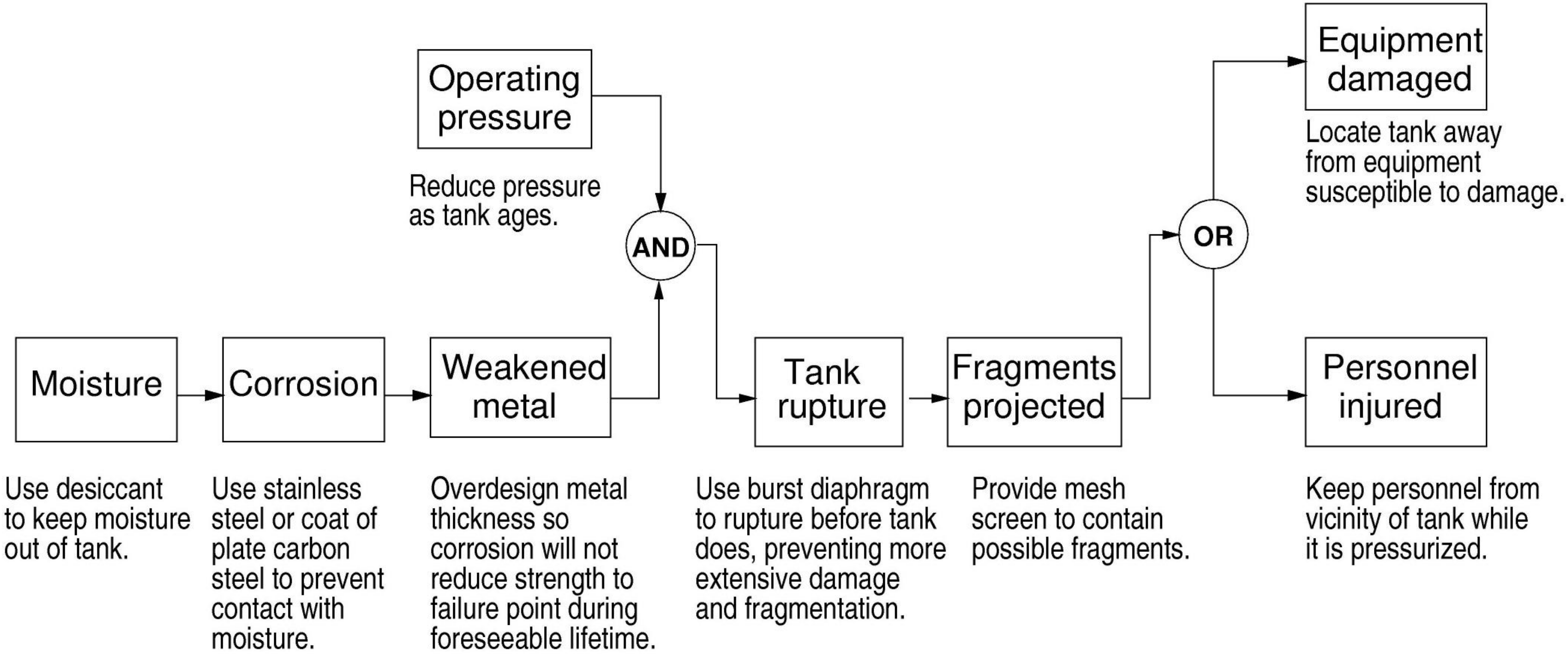
Systems Thinking & Systems Theory

(Leveson, 2012)

STAMP is an Accident Model

- What is an accident model?
- What is STAMP?
- What is STPA?

Accident model: Chain-of-events example



Operating pressure

Reduce pressure as tank ages.

AND

Equipment damaged

Locate tank away from equipment susceptible to damage.

OR

Moisture

Use desiccant to keep moisture out of tank.

Corrosion

Use stainless steel or coat of plate carbon steel to prevent contact with moisture.

Weakened metal

Overdesign metal thickness so corrosion will not reduce strength to failure point during foreseeable lifetime.

Tank rupture

Use burst diaphragm to rupture before tank does, preventing more extensive damage and fragmentation.

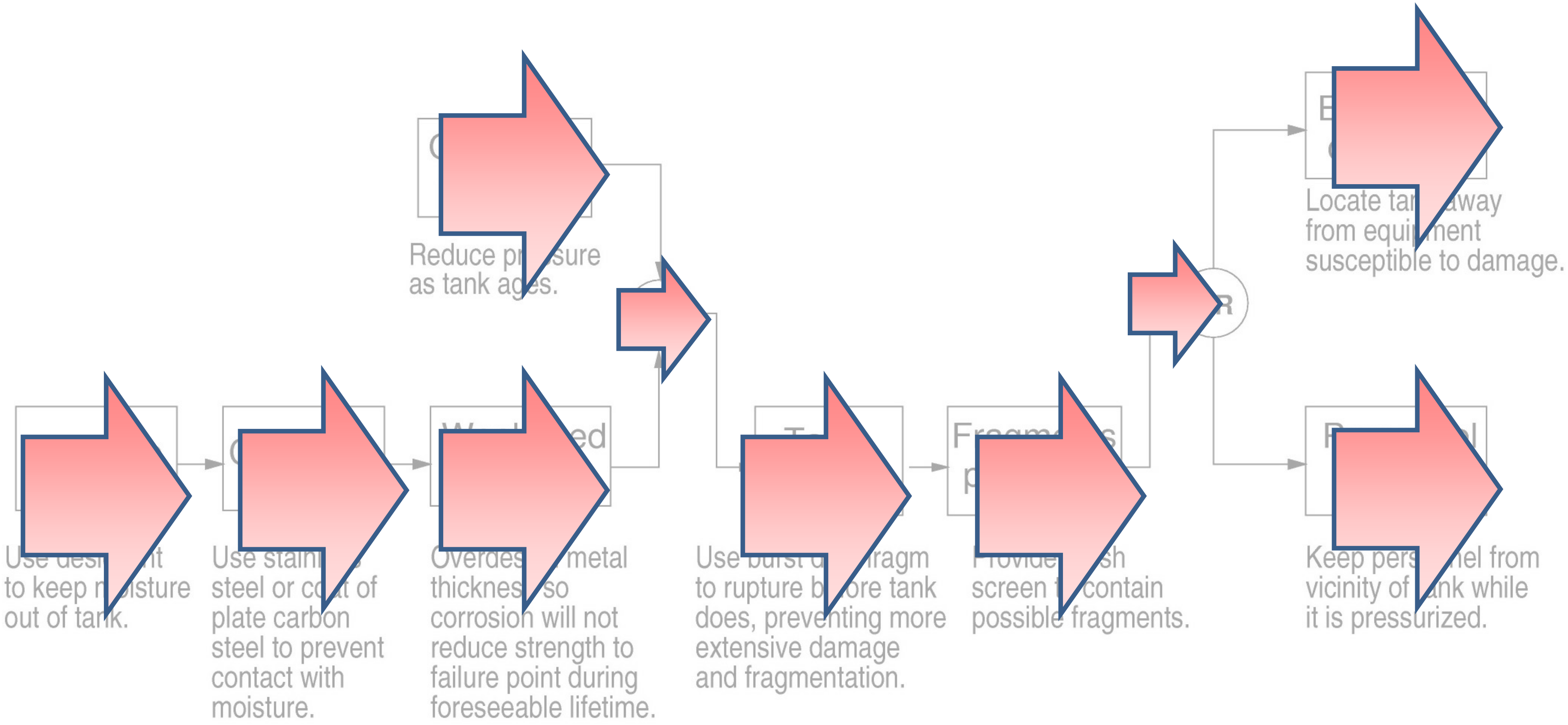
Fragments projected

Provide mesh screen to contain possible fragments.

Personnel injured

Keep personnel from vicinity of tank while it is pressurized.

Accident model: Chain-of-events example



Reduce pr sure as tank ag s.

Use desi nt to keep m isture out of tank.

Use stain s steel or co t of plate carbon steel to prevent contact with moisture.

Overdes metal thicknes so corrosion will not reduce strength to failure point during foreseeable lifetime.

Use burst d ragm to rupture b ore tank does, preventing more extensive damage and fragmentation.

Provide sh screen t contain possible fragments.

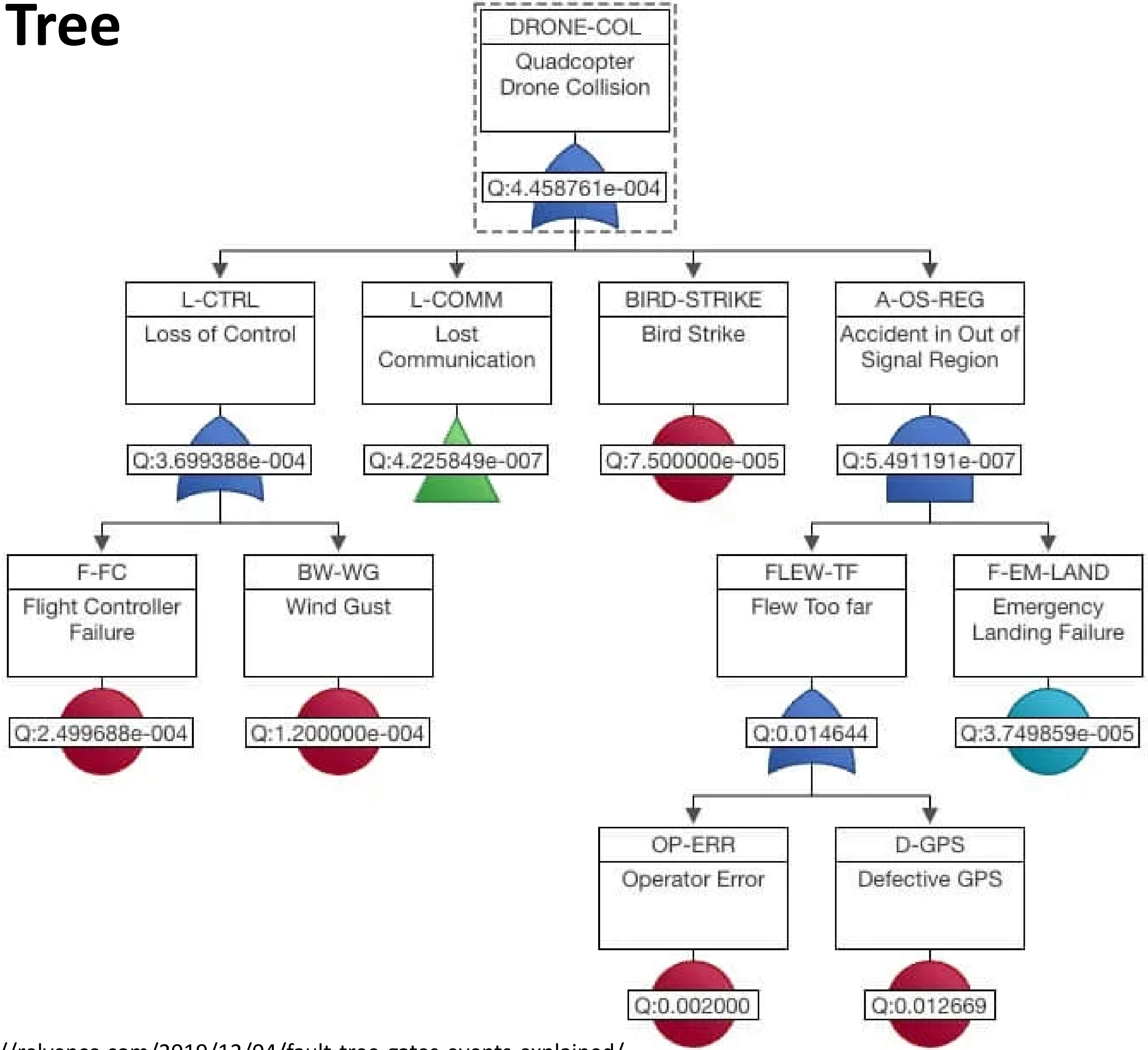
Locate tan away from equi ment susceptible to damage.

Keep pers nel from vicinity of nk while it is pressurized.

Emphasizes redundancy, fault propagation

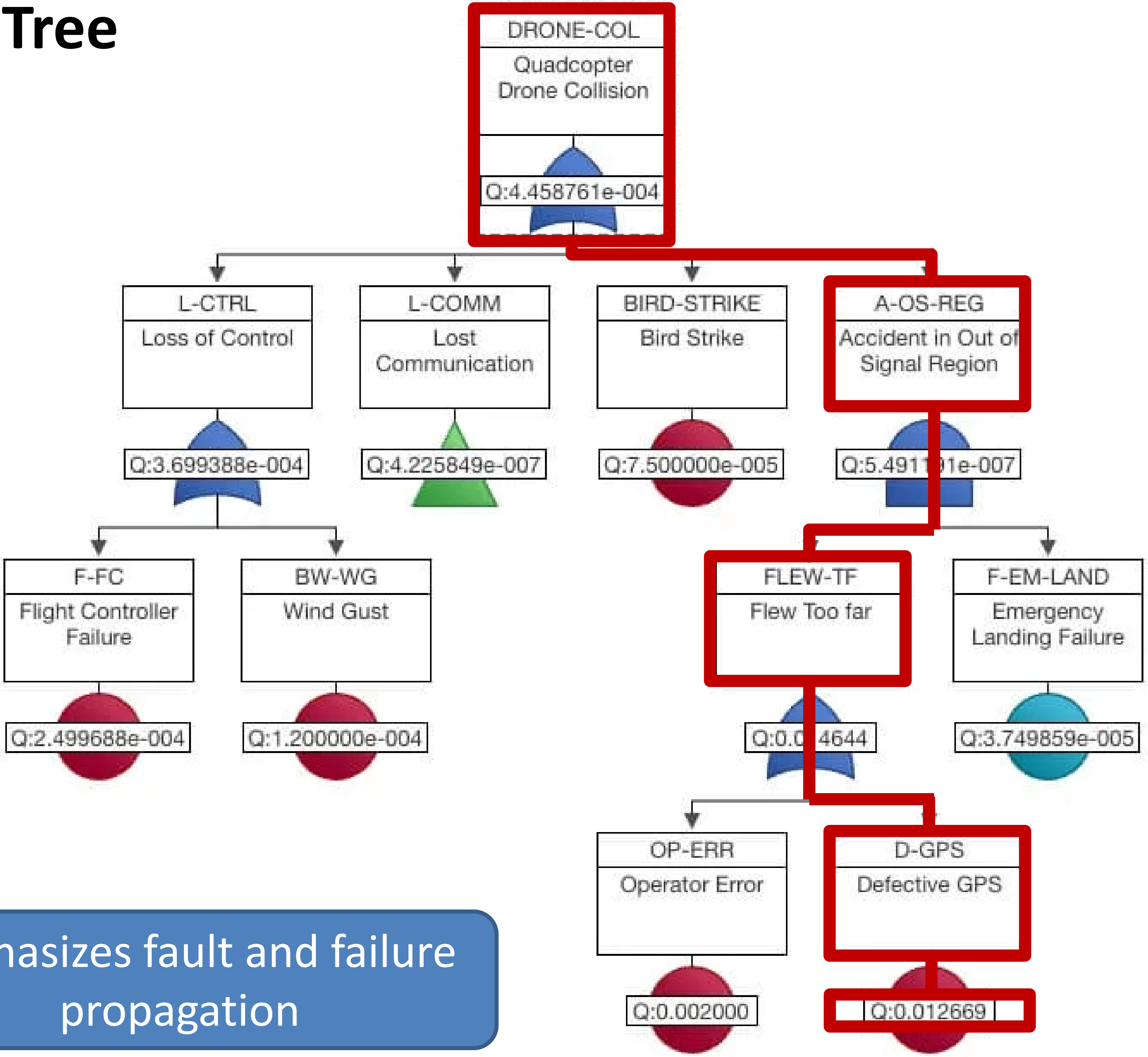
Often used in Fault Tree Analysis

Fault Tree



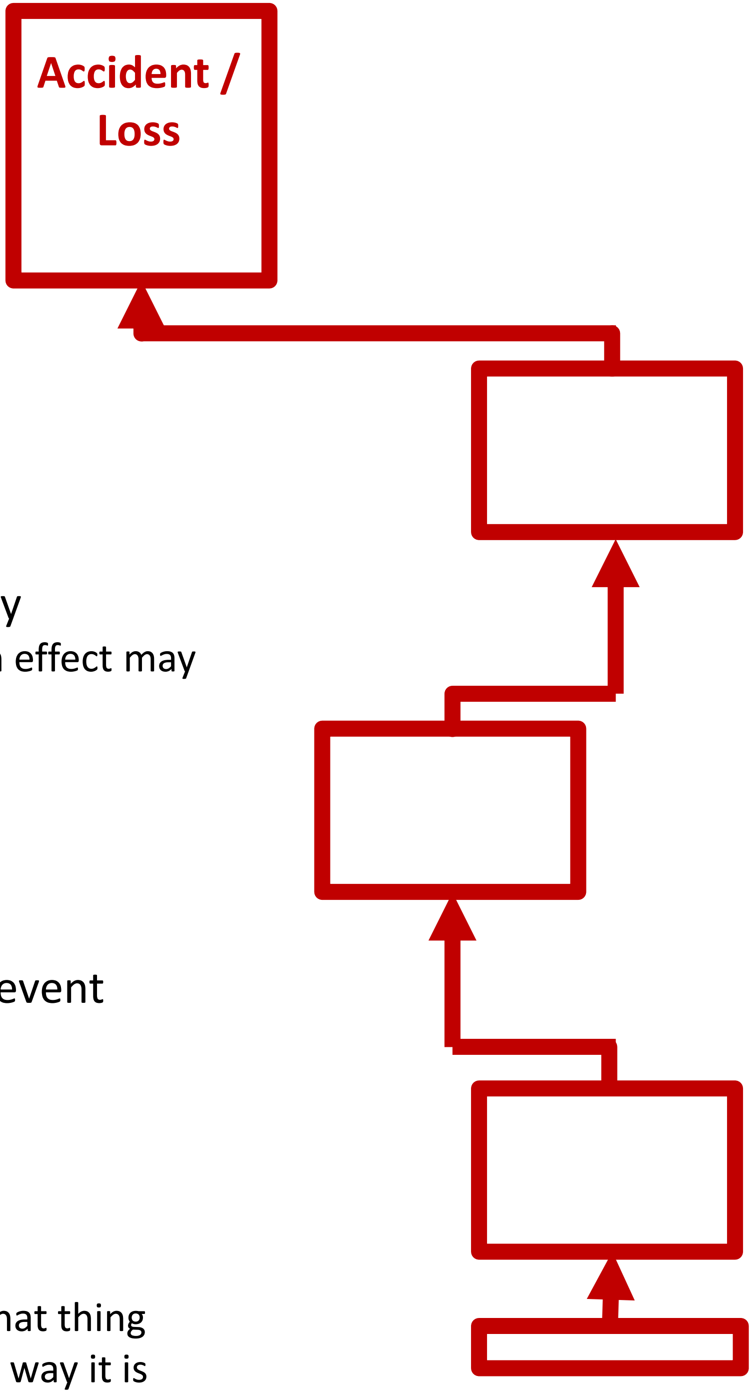
Example: https://relyence.com/2019/12/04/fault-tree-gates-events-explained/

Fault Tree



Emphasizes fault and failure propagation

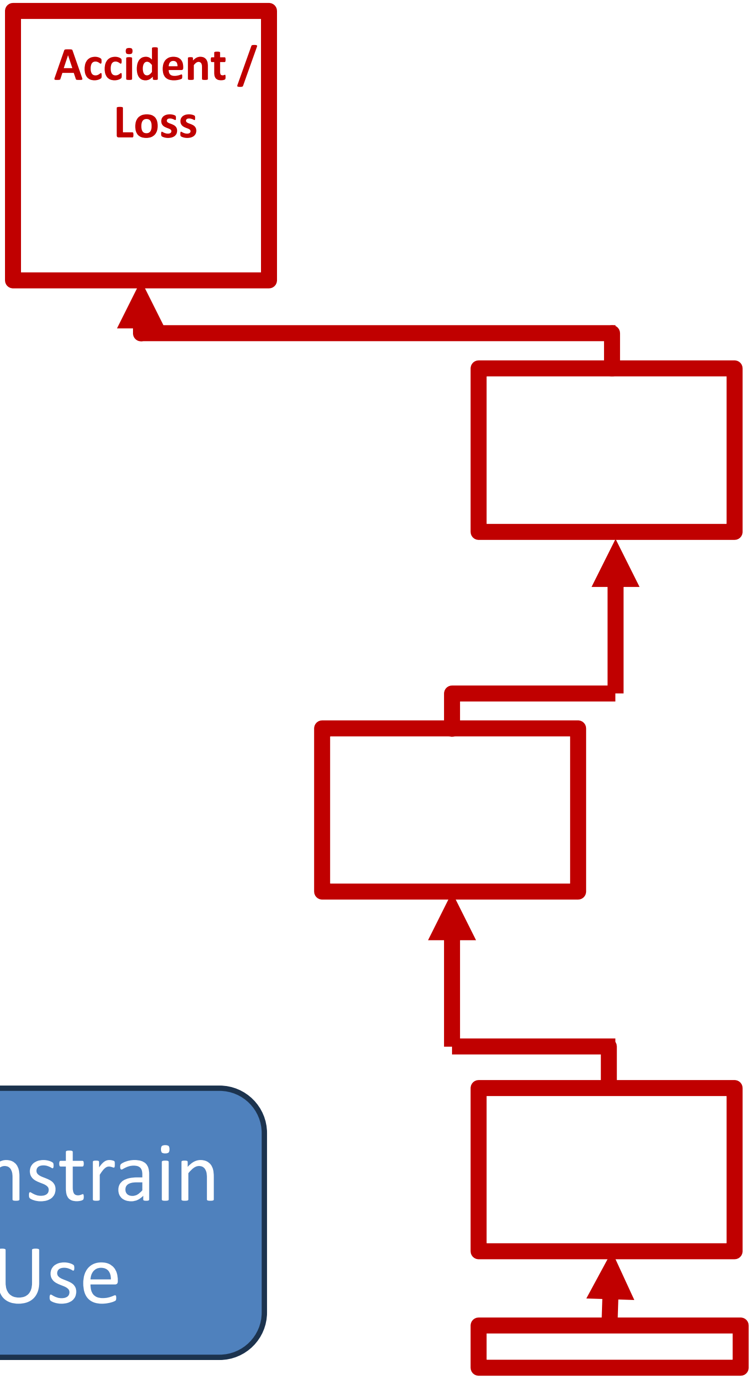
Chain of Failure Events Accident Model



Properties of this accident model:

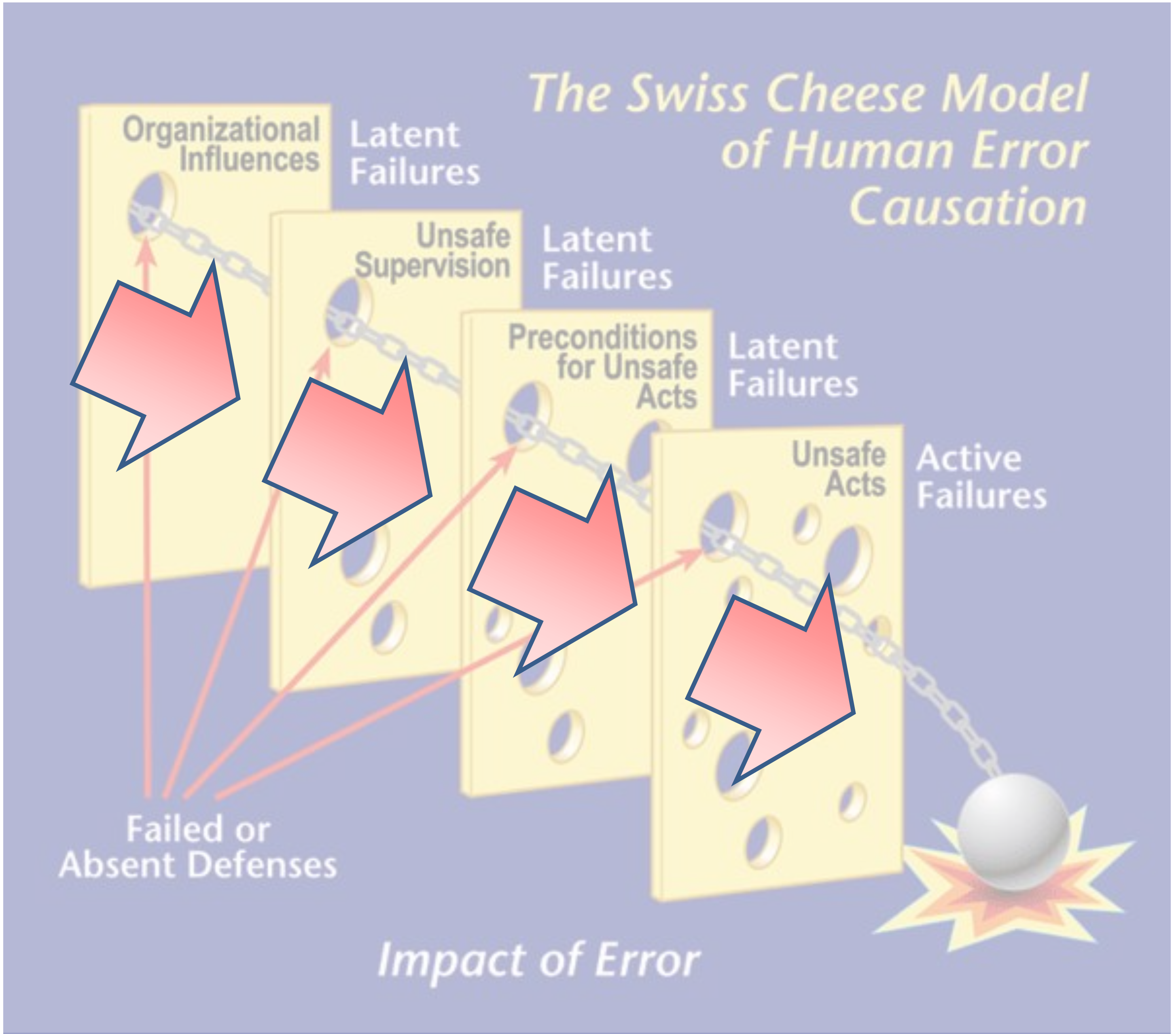
- Events are defined as faults / failures
  - Deviations from intended/specified behavior
  - Excludes intended / specified behaviors
- Events are binary
  - Must resolve as true or false
- Event sequence usually modeled deterministically
  - Influences that influence but do not guarantee an effect may not be modeled
- One-to-one or many-to-one propagation
  - Not many-to-many
- Linear propagation in one direction
- Loops (circular causality) not modeled
- Events may not be caused by the same identical event previously
- Does not model reasoning and decision-making
  - E.g., beliefs, past experiences
- Models events, not the reasons for them
  - Often assumes the cause is random
  - Not intended to explain *why* a person would do that thing
  - Not intended to explain *why* a design is made the way it is

Chain of Failure Events Accident Model



Accident Models Constrain the Methods We Use

Swiss Cheese Accident Model

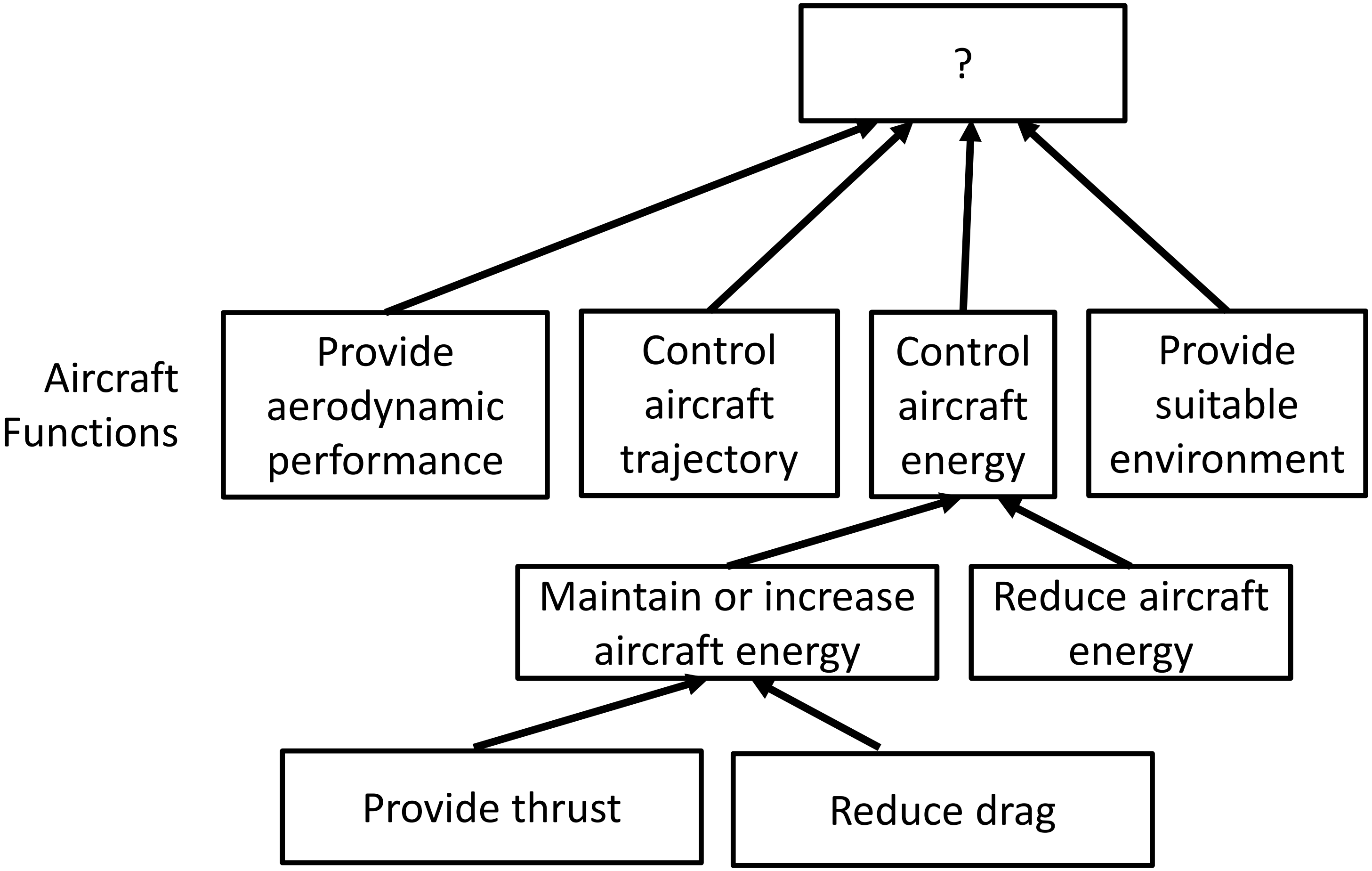


The Swiss Cheese Model of Human Error Causation
Organizational Influences
Latent Failures
Unsafe Supervision
Latent Failures
Preconditions for Unsafe Acts
Latent Failures
Unsafe Acts
Active Failures
Failed or Absent Defenses
Impact of Error

Image from: http://www.fireengineering.com/articles/print/volume-163/issue-3/features/managing-fireground-errors.html

System Models

Example: Functional Decomposition

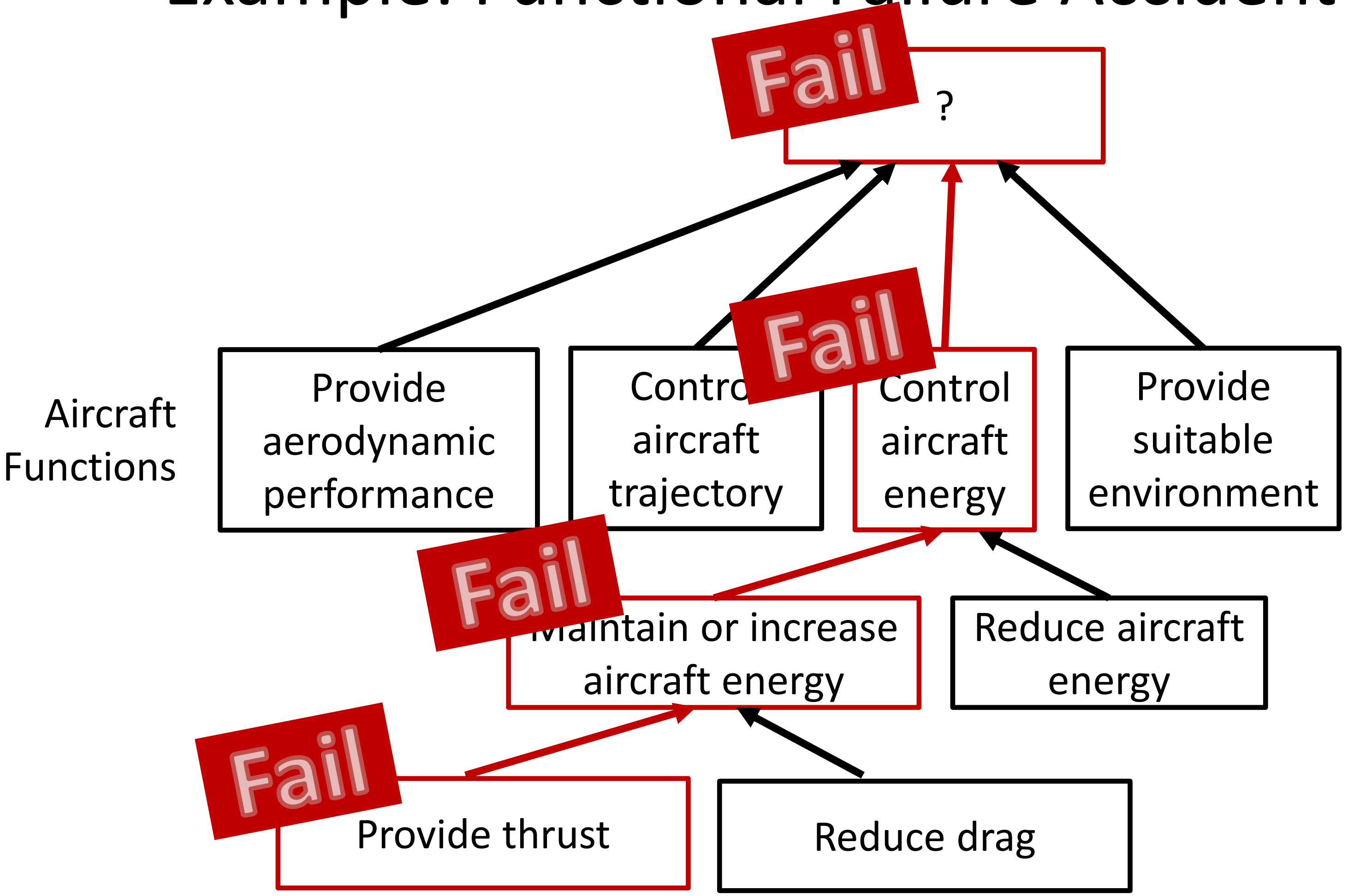

?
Aircraft Functions
Provide aerodynamic performance
Control aircraft trajectory
Control aircraft energy
Provide suitable environment
Maintain or increase aircraft energy
Reduce aircraft energy
Provide thrust
Reduce drag

Emphasizes individual functions

Used in Functional Hazard Assessment (FHA), PFMEA

Accident Models

Example: Functional Failure Accident Model

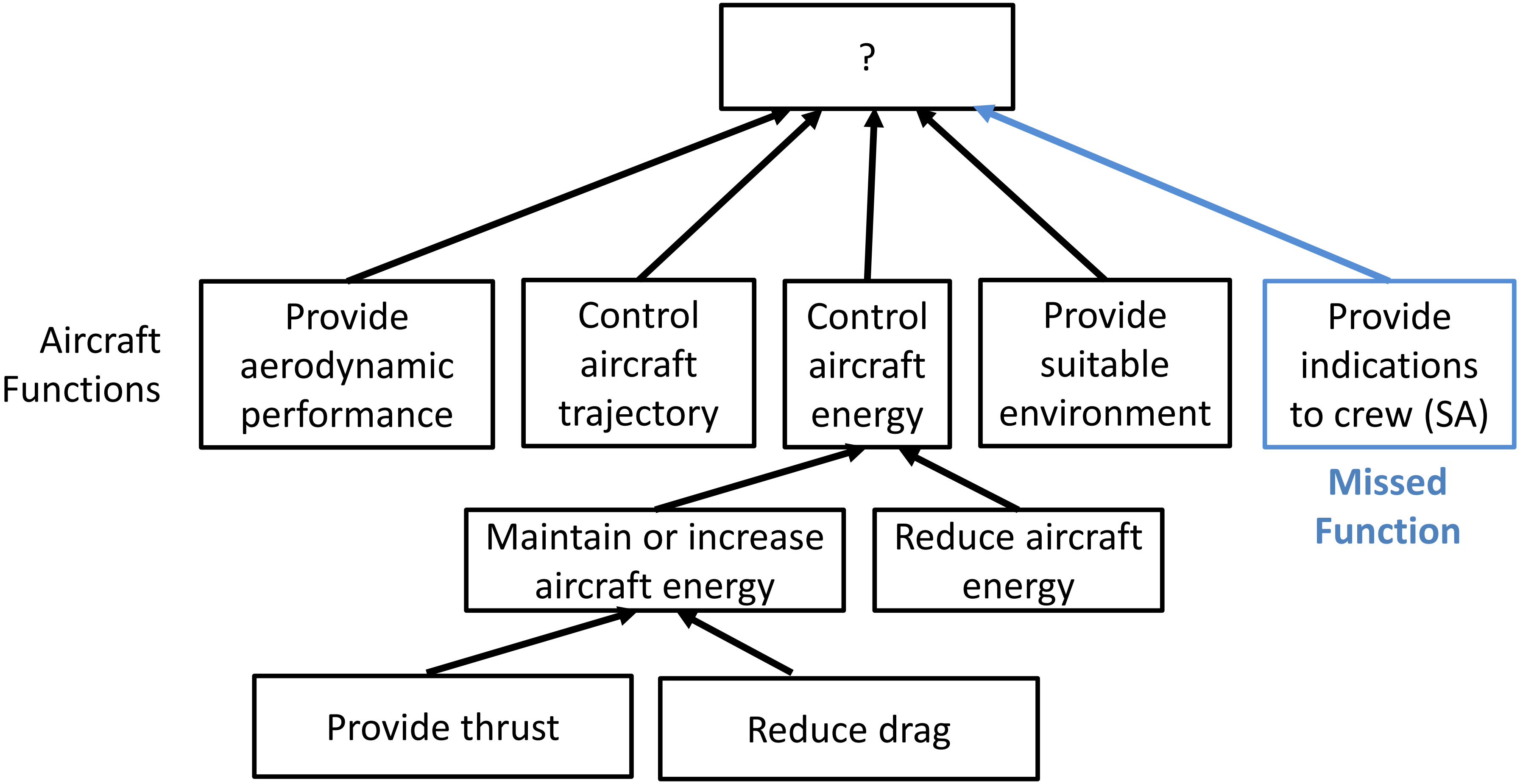

Fail
?
Aircraft Functions
Provide aerodynamic performance
Contro aircraft trajectory
Fail
Control aircraft energy
Provide suitable environment
Fail
Maintain or increase aircraft energy
Reduce aircraft energy
Fail
Provide thrust
Reduce drag

Emphasizes individual functions

Used in Functional Hazard Assessment (FHA), PFMEA

System Models

Example: Functional Decomposition

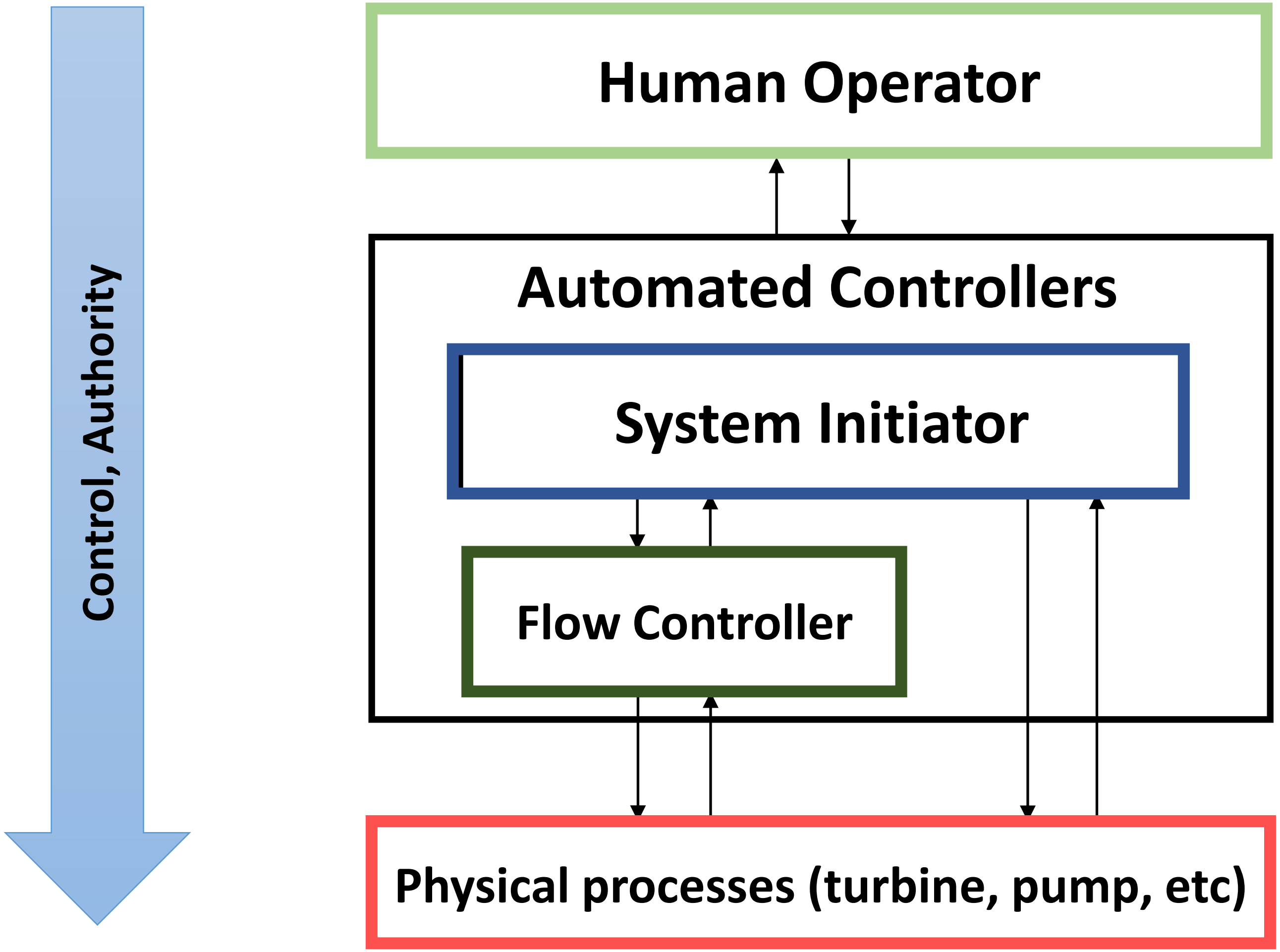

?
Aircraft Functions
Provide aerodynamic performance
Control aircraft trajectory
Control aircraft energy
Provide suitable environment
Provide indications to crew (SA)
Missed Function
Maintain or increase aircraft energy
Reduce aircraft energy
Provide thrust
Reduce drag

Doesn’t help identify missing functions that are needed or functions that may be unsafe as designed

- What is an accident model?
- What is STAMP?
- What is STPA?

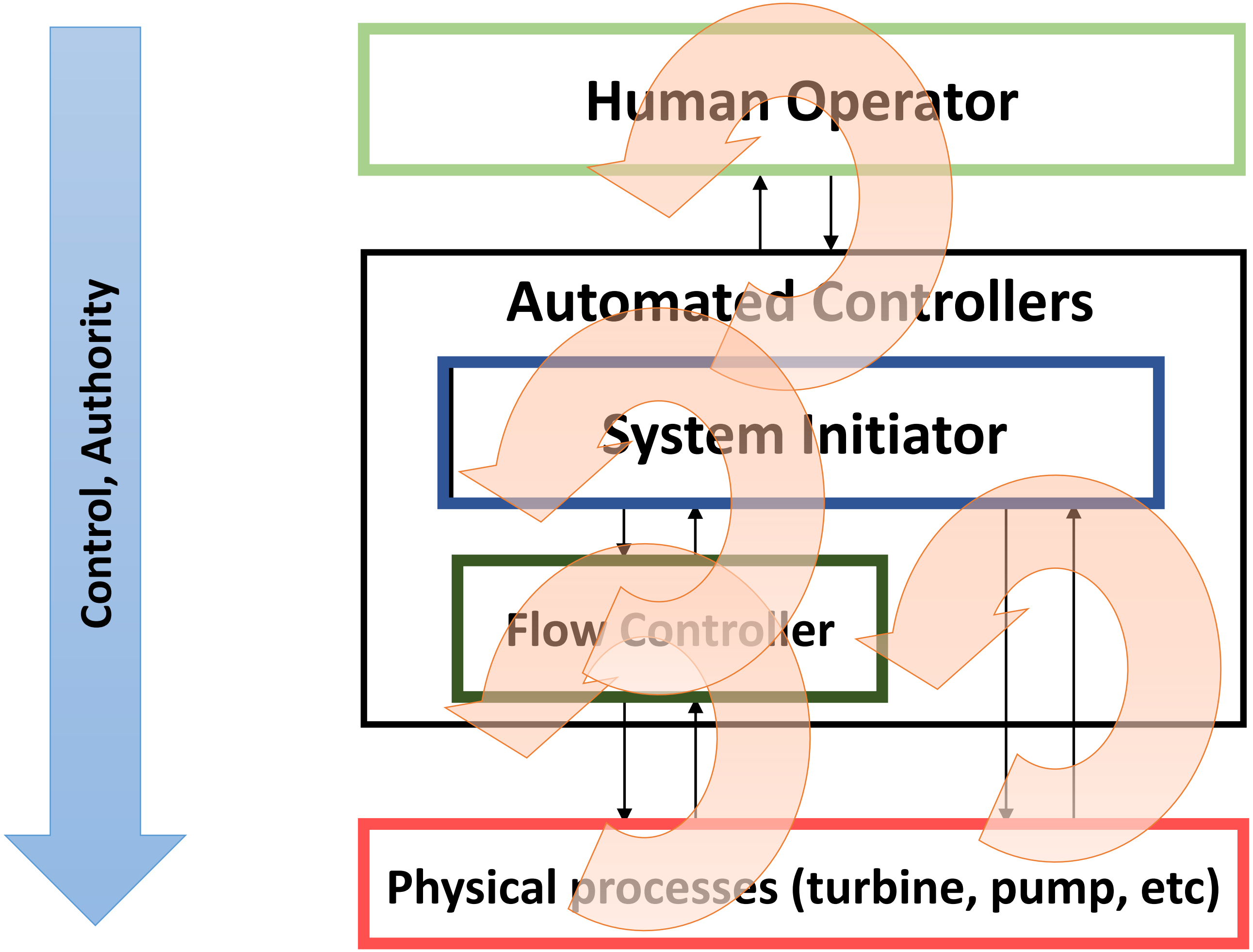
System Models

Example: Control Structure



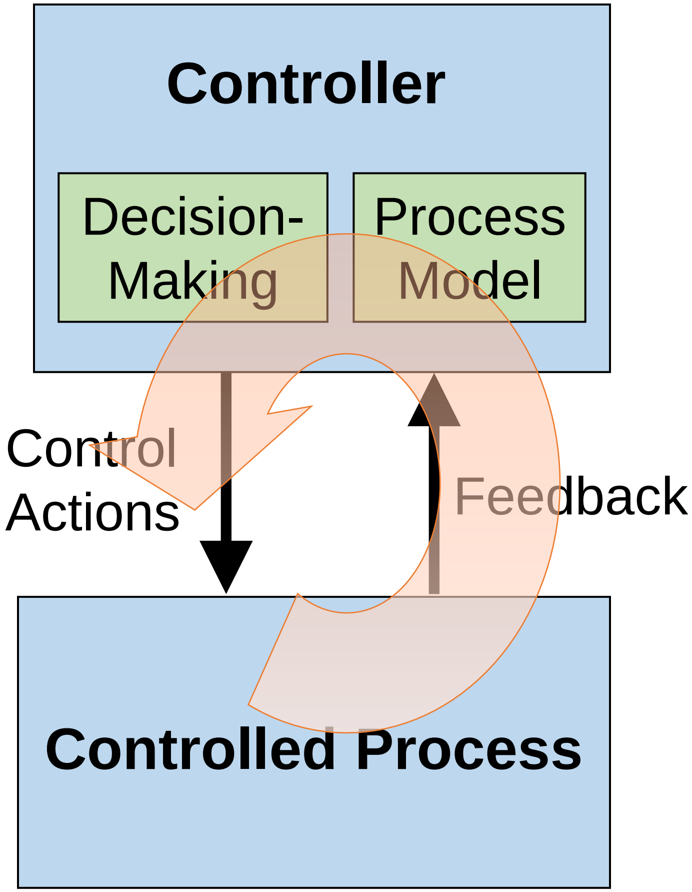
Emphasizes control relationships
Used in STAMP / STPA

Accident Causation in STAMP



Emphasizes control relationships
Used in STAMP / STPA

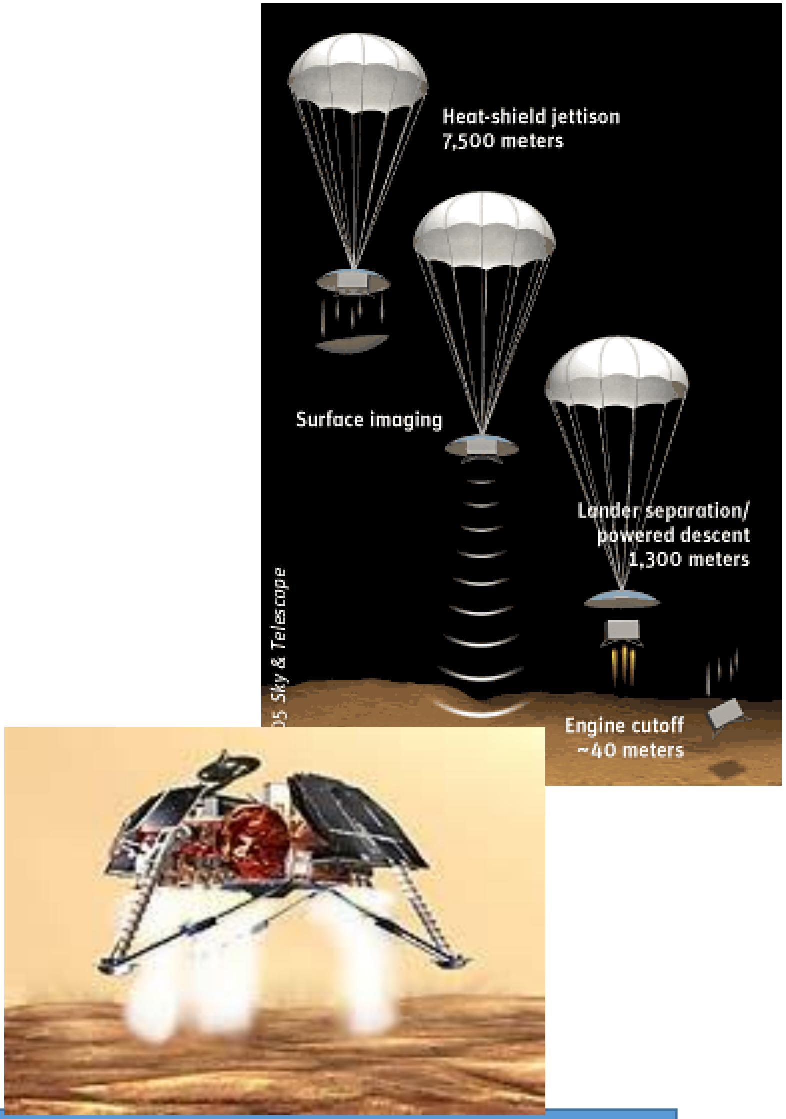
Basic control loop



- **Control actions** are provided to affect a controlled process
- **Feedback** may be used to monitor the process
- **Process model** (beliefs) formed based on feedback and other information
- **Decision-making** determines appropriate control actions given current beliefs

Mars Polar Lander

- During descent to Mars, legs deployed (as planned)
- Footpad sensors detected vibration (within design spec)
- Momentary signal sent to computer (as required)
- Computer shut down the descent engines (as specified)
- The vehicle free-fell, fell to surface at 50 mph (80 kph), destroyed



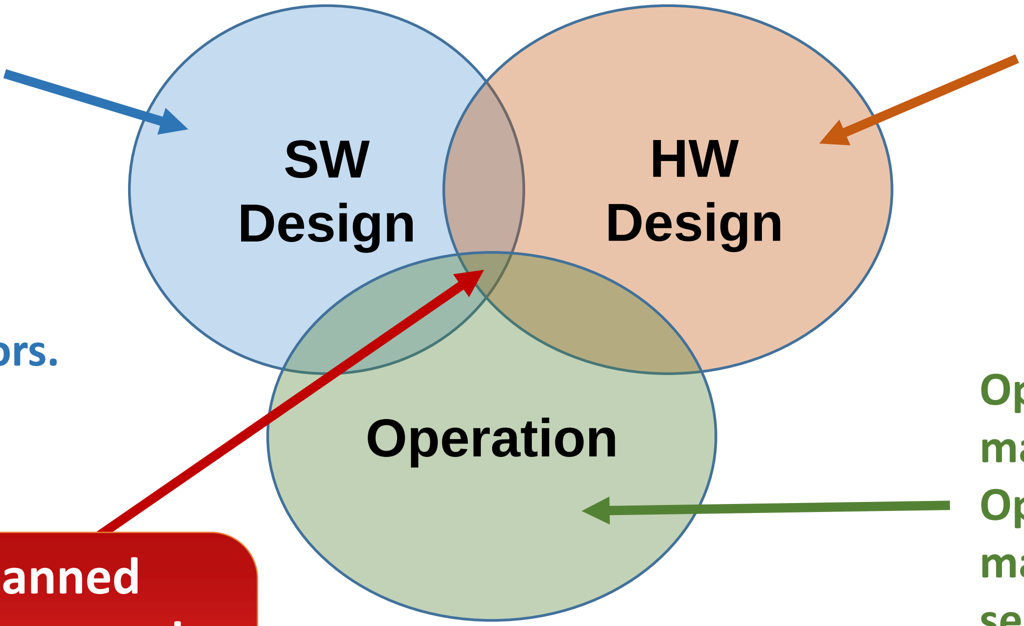
All components performed exactly as designed!
No single component failed!

John Thomas, 2019

MPL: How was this overlooked?

SW worked as designed! Operation did not match any defined SW component failure behaviors.

HW worked as designed! Operation did not match any defined HW component failure behaviors.



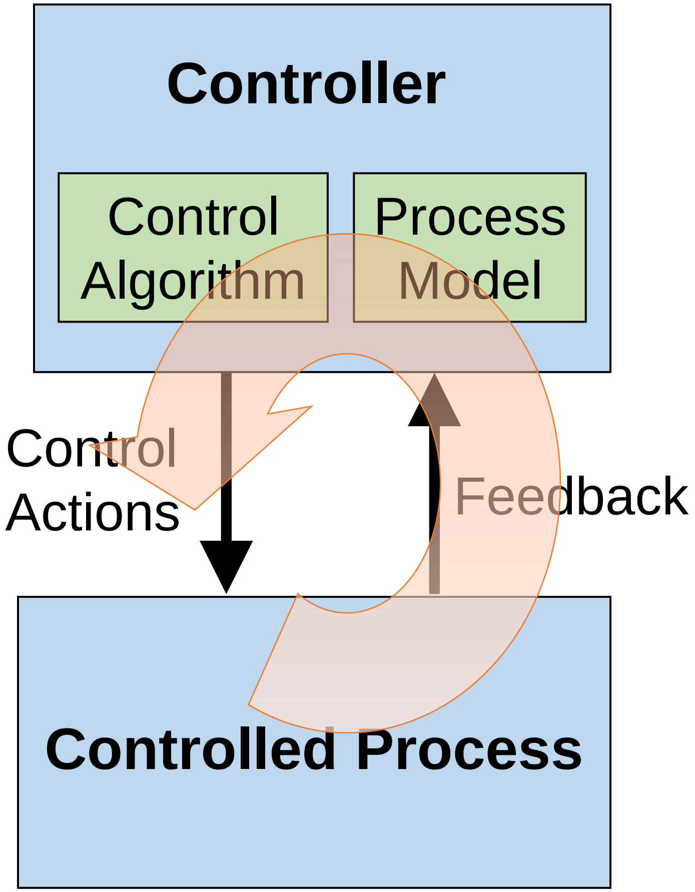
Operation sequence matched design intent. Operation sequence did not match any defined sequences of failure.

New, unplanned interaction emerged among whole ***system*** of components!

Easy to overlook the system problem by looking at individual component failures

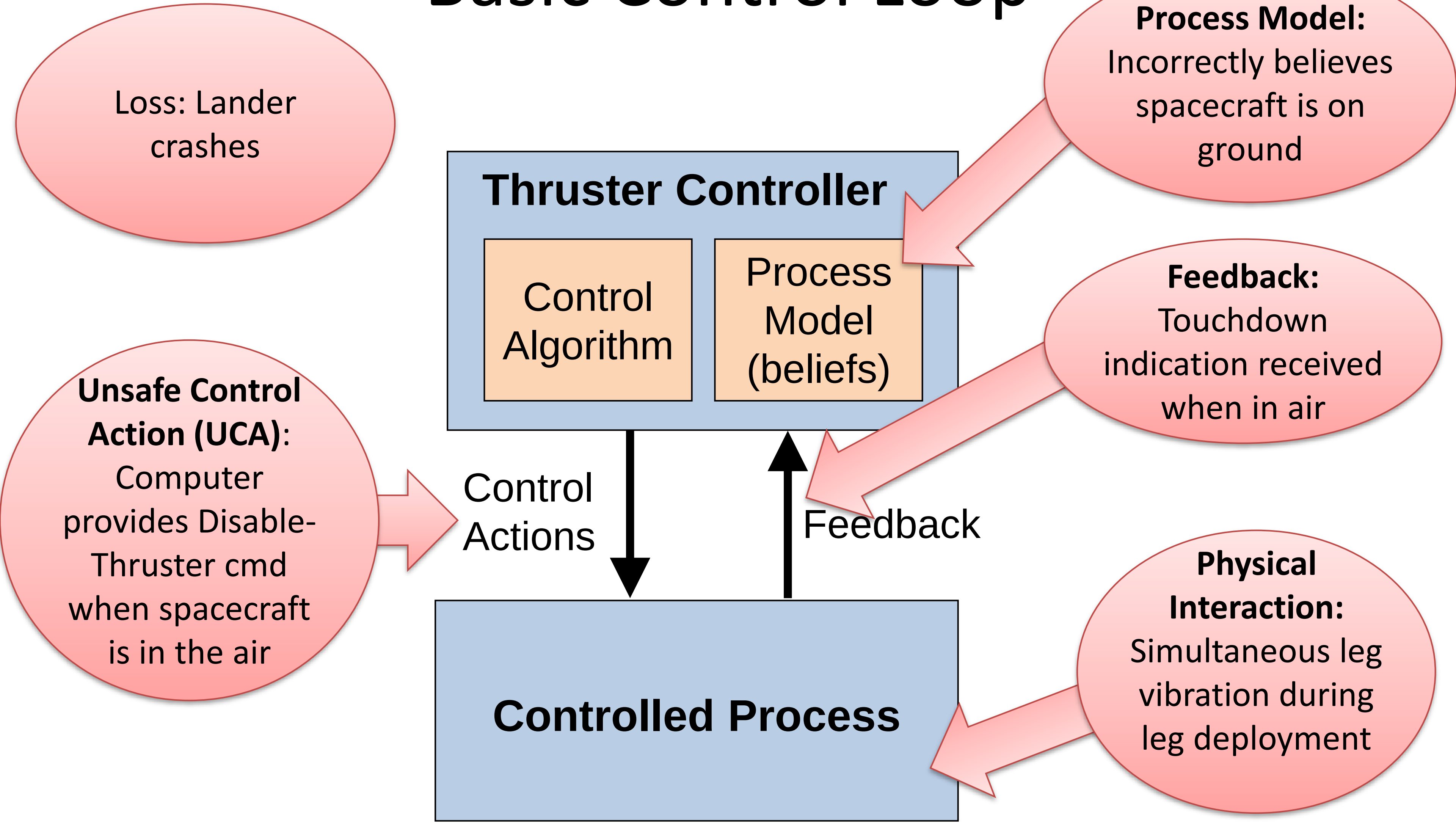
John Thomas, 2019

Basic control loop



- **Control actions** are provided to affect a controlled process
- **Feedback** may be used to monitor the process
- **Process model** (beliefs) formed based on feedback and other information
- **Control algorithm** determines appropriate control actions given current beliefs

Basic Control Loop



Loss: Lander crashes

Thruster Controller

Control Algorithm

Process Model (beliefs)

Control Actions

Feedback

Controlled Process

Process Model: Incorrectly believes spacecraft is on ground

Feedback: Touchdown indication received when in air

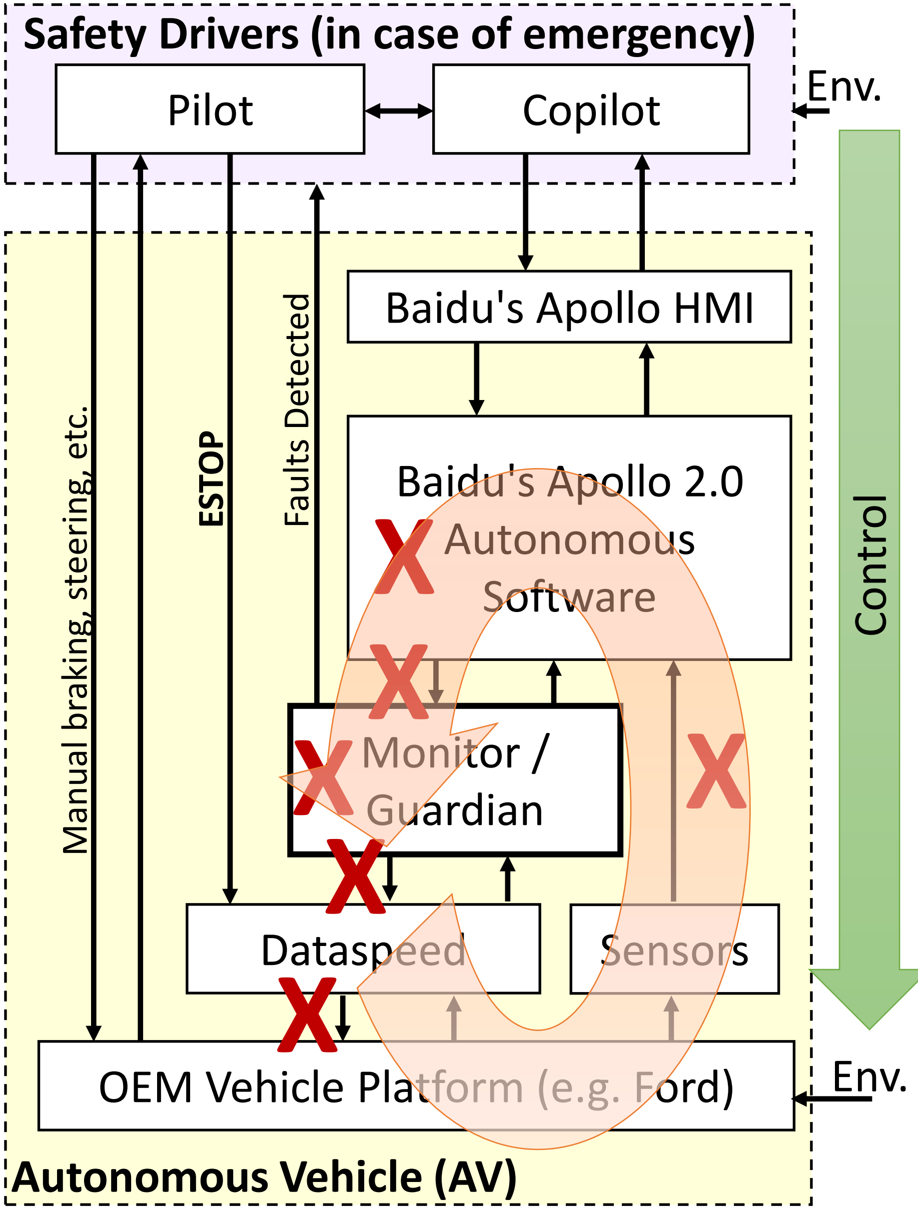
Unsafe Control Action (UCA): Computer provides Disable-Thruster cmd when spacecraft is in the air

Physical Interaction: Simultaneous leg vibration during leg deployment

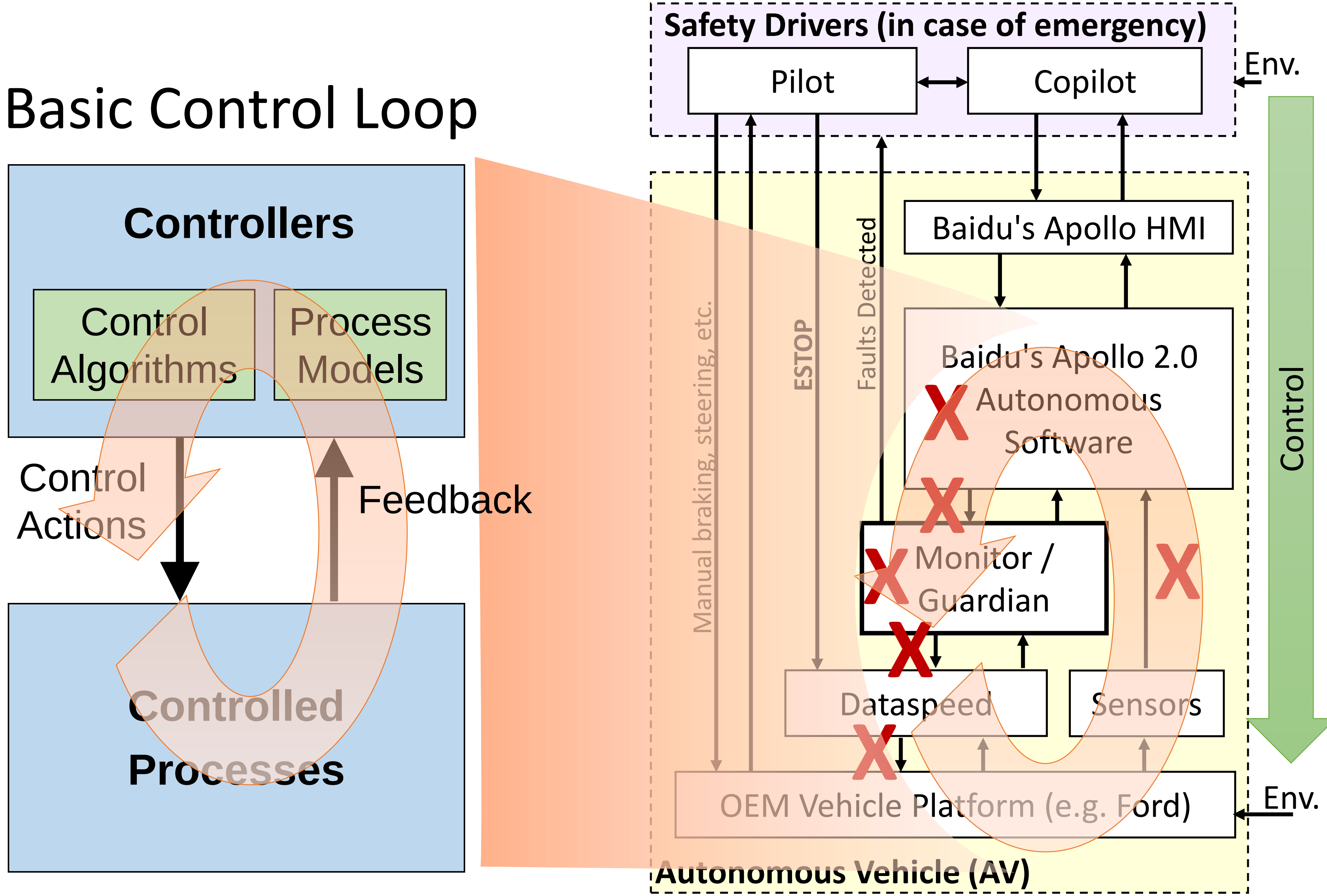
This framework works with or without component failures!

A Systems Approach to Safety

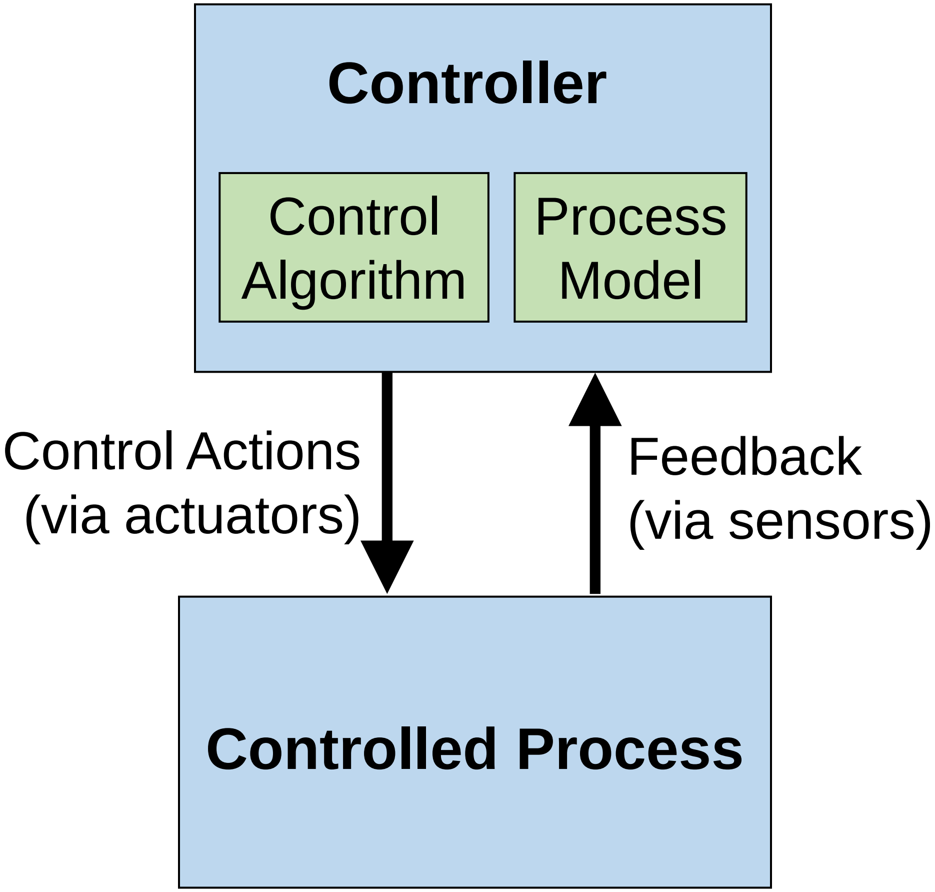
Treat safety as a **control problem**


Safety Drivers (in case of emergency)
Pilot
Copilot
Env.
Manual braking, steering, etc.
ESTOP
Faults Detected
Baidu's Apollo HMI
Baidu's Apollo 2.0 Autonomous Software
Monitor / Guardian
Dataspeed
Sensors
OEM Vehicle Platform (e.g. Ford)
Env.
Control
X
X
X
X
X
X
Autonomous Vehicle (AV)

A Systems Approach to Safety

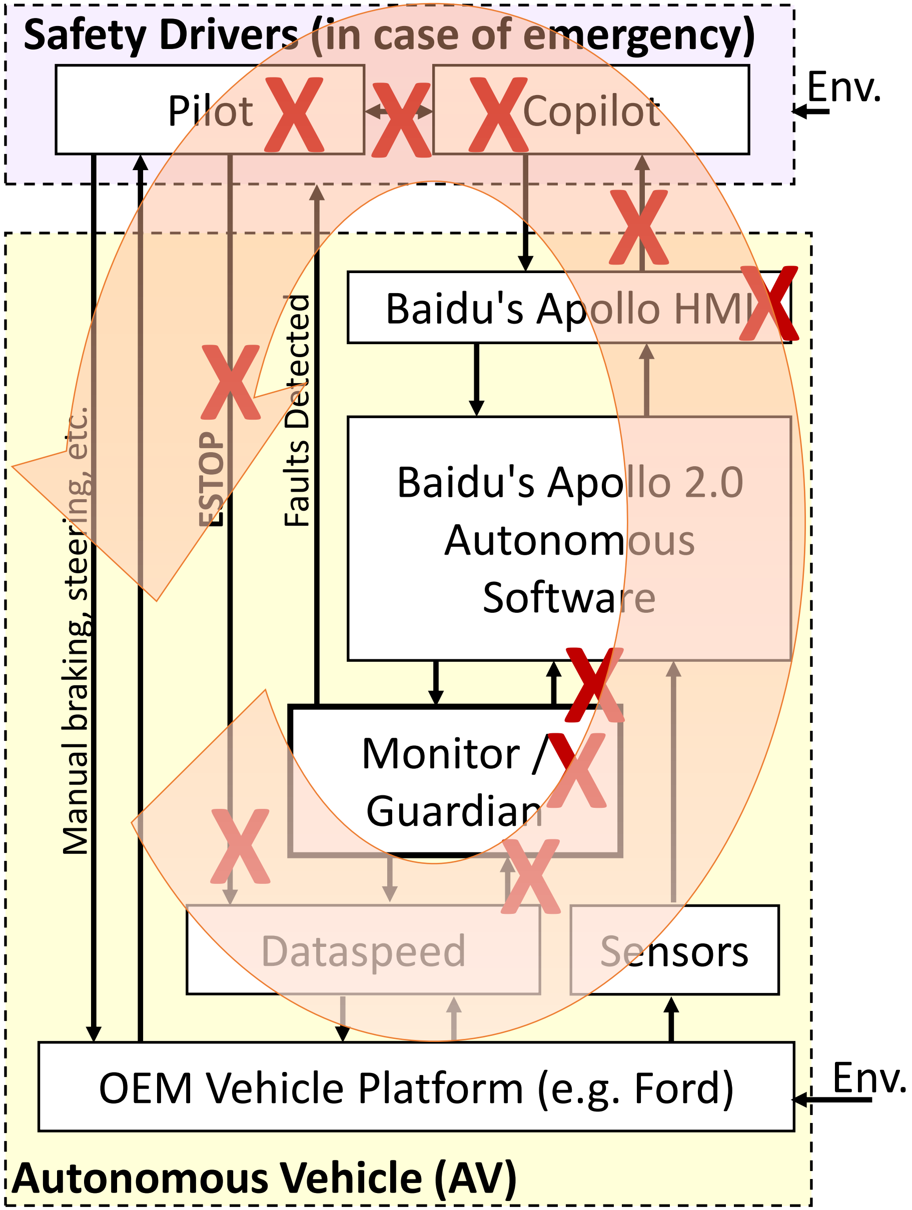


Basic Control Loop



- **Control actions** are provided to affect a controlled process
- **Feedback** may be used to monitor the process
- **Process model** (beliefs) formed based on feedback and other information
- **Control algorithm** determines appropriate control actions given current beliefs

A Systems Approach to Safety

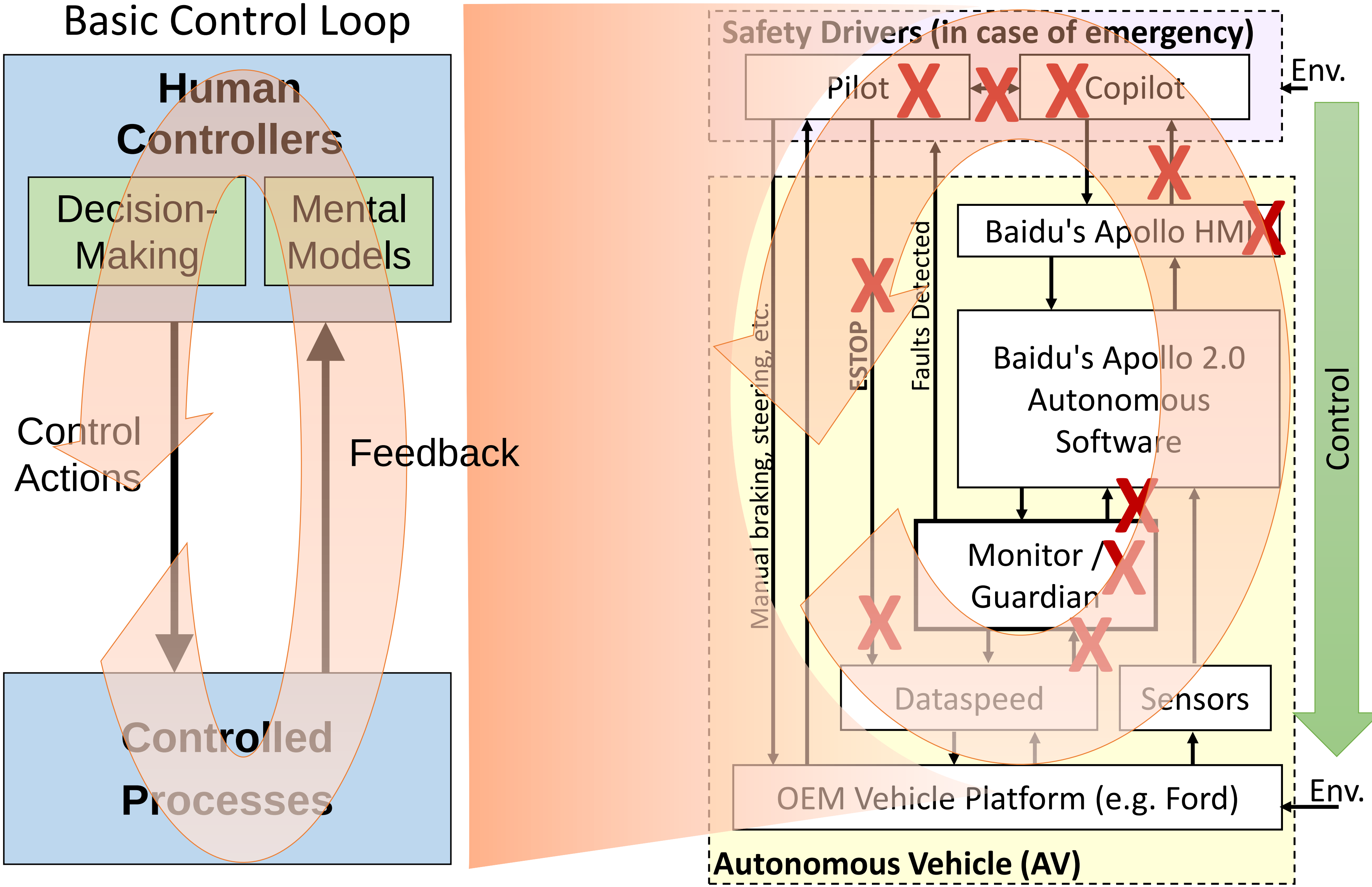


Safety Drivers (in case of emergency)
Pilot
Copilot
Env.
Baidu's Apollo HMI
Baidu's Apollo 2.0 Autonomous Software
Monitor / Guardian
Dataspeed
Sensors
OEM Vehicle Platform (e.g. Ford)
Env.
Manual braking, steering, etc.
ESTOP
Faults Detected
Autonomous Vehicle (AV)

HMI = Human-Machine Interface

ESTOP = Emergency Stop

A Systems Approach to Safety

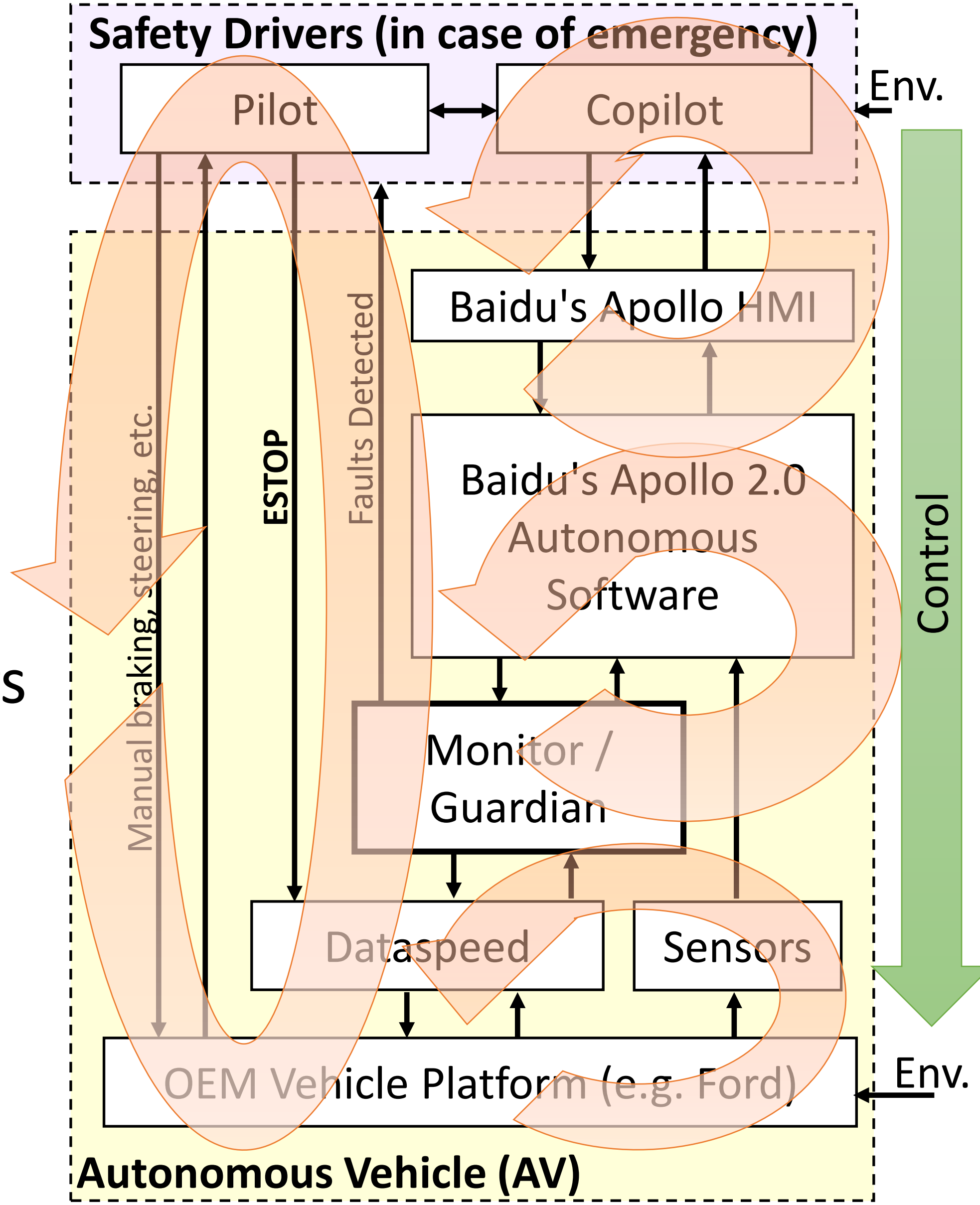


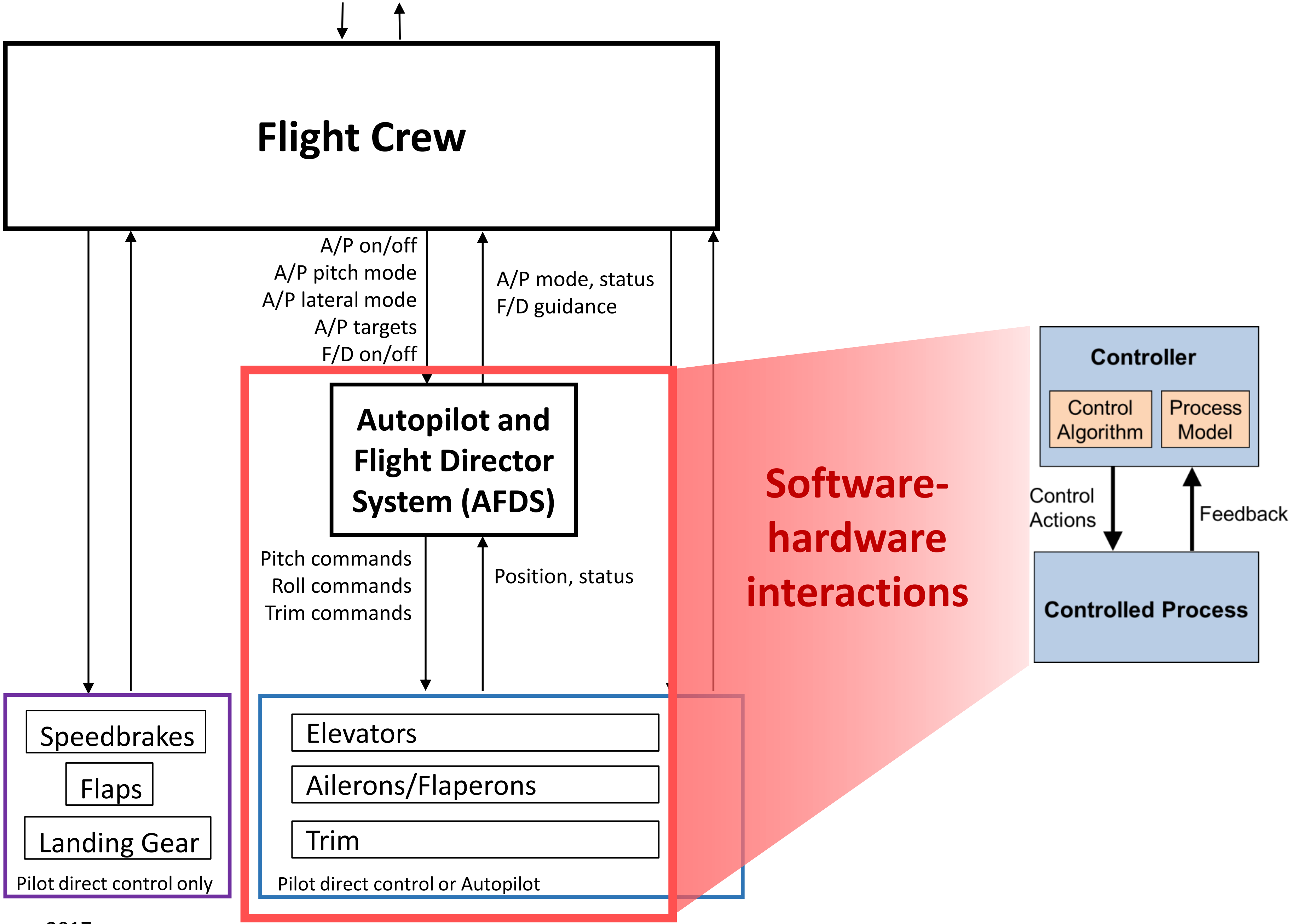
A Systems Approach to Safety

Treat accidents as a **control problem**

Works well to anticipate:

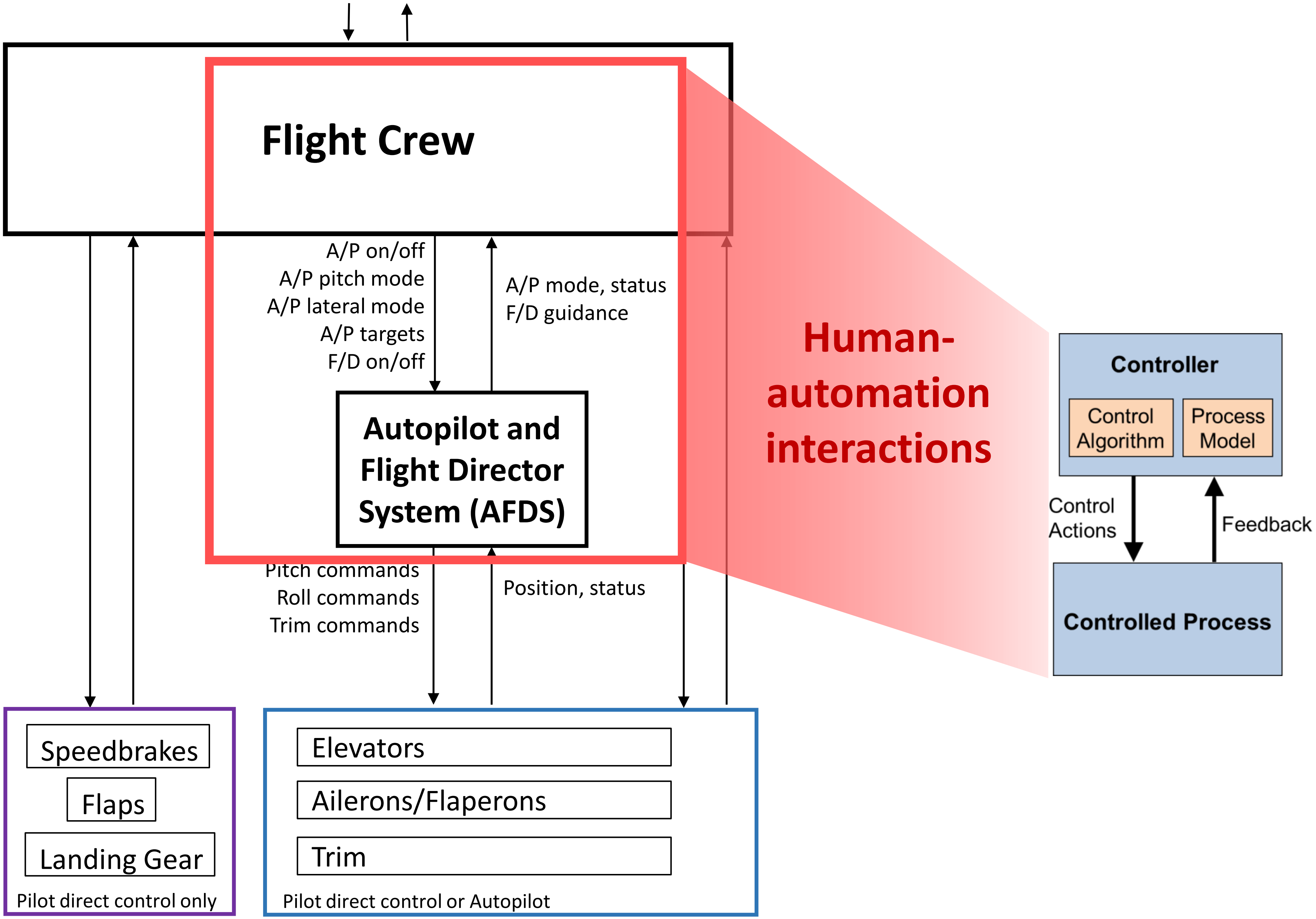
- **Interactions** between new functions and features
- Complex **Automated** behaviors
- Complex **Human** behaviors
- **“unknown unknowns”** in engineering
- Engineering **Assumptions**


Safety Drivers (in case of emergency)
Pilot
Copilot
Env.
Control
Baidu's Apollo HMI
Baidu's Apollo 2.0 Autonomous Software
Manual braking, steering, etc.
ESTOP
Faults Detected
Monitor / Guardian
Dataspeed
Sensors
OEM Vehicle Platform (e.g. Ford)
Env.
Autonomous Vehicle (AV)



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
Pilot direct control only
Elevators
Ailerons/Flaperons
Trim
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

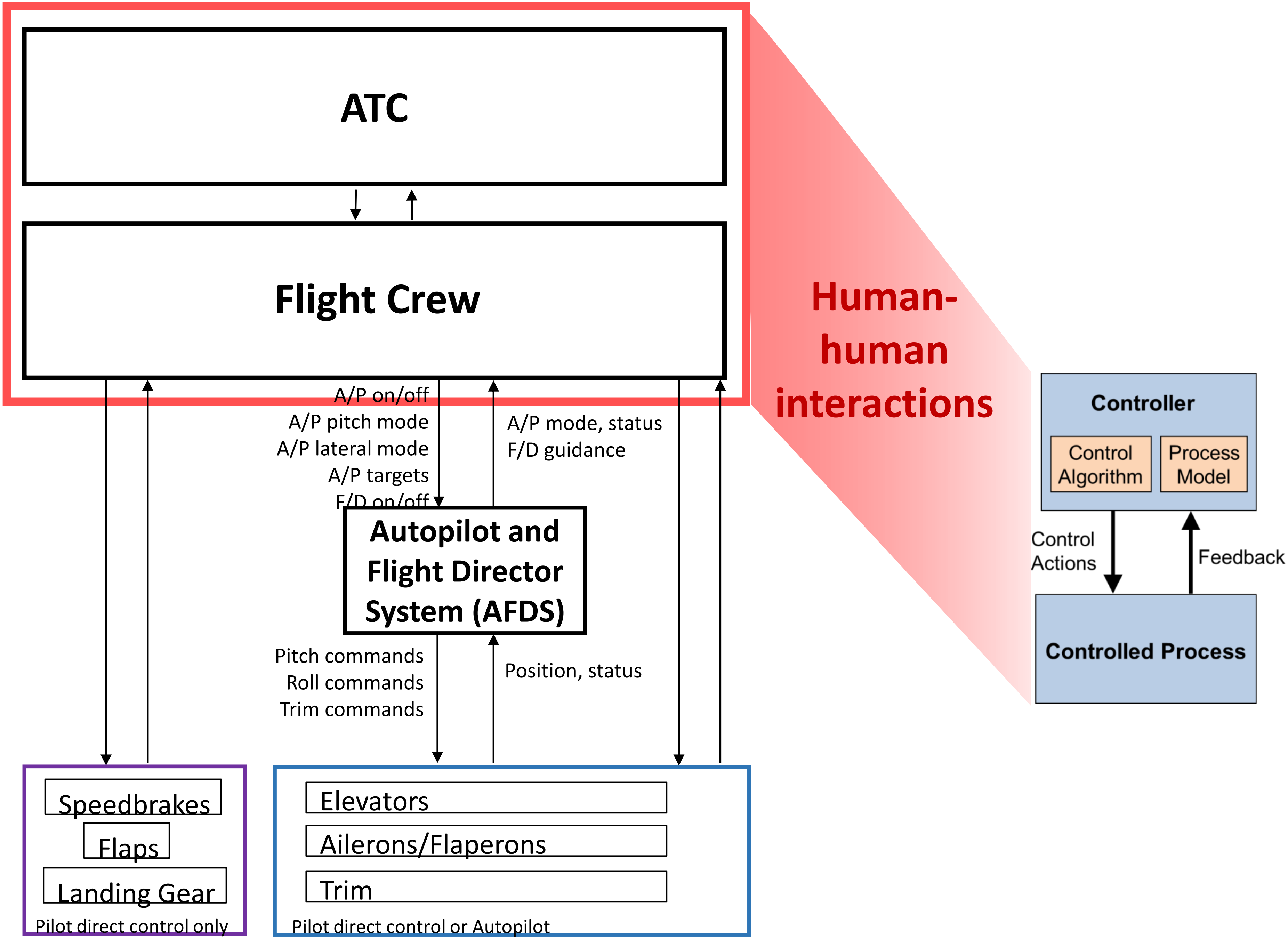
Speedbrakes
Flaps
Landing Gear
Pilot direct control only

Elevators
Ailerons/Flaperons
Trim
Pilot direct control or Autopilot

Human-automation interactions

Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process

Thomas, 2017



ATC

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
Pilot direct control only

Elevators
Ailerons/Flaperons
Trim
Pilot direct control or Autopilot

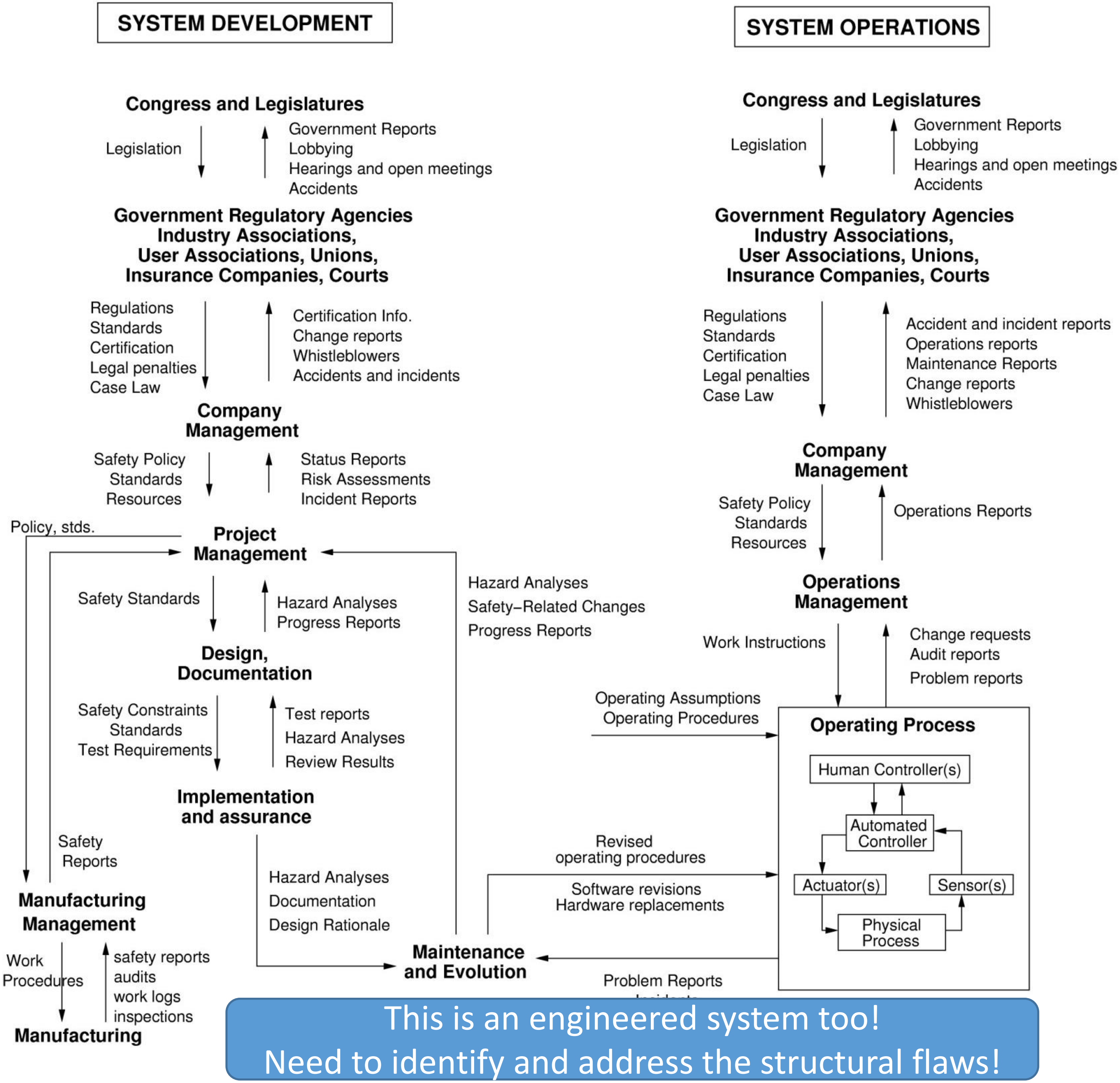
Human-human interactions

Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process

Thomas, 2017

Example Safety Control Structure

Control, Authority



SYSTEM DEVELOPMENT

Congress and Legislatures

Legislation

Government Reports
Lobbying
Hearings and open meetings
Accidents

Government Regulatory Agencies
Industry Associations,
User Associations, Unions,
Insurance Companies, Courts

Regulations
Standards
Certification
Legal penalties
Case Law

Certification Info.
Change reports
Whistleblowers
Accidents and incidents

Company Management

Safety Policy
Standards
Resources

Status Reports
Risk Assessments
Incident Reports

Policy, stds.

Project Management

Safety Standards

Hazard Analyses
Progress Reports

Hazard Analyses
Safety–Related Changes
Progress Reports

Design, Documentation

Safety Constraints
Standards
Test Requirements

Test reports
Hazard Analyses
Review Results

Implementation and assurance

Safety Reports

Hazard Analyses
Documentation
Design Rationale

Manufacturing Management

Work Procedures

safety reports
audits
work logs
inspections

Manufacturing

Maintenance and Evolution

SYSTEM OPERATIONS

Congress and Legislatures

Legislation

Government Reports
Lobbying
Hearings and open meetings
Accidents

Government Regulatory Agencies
Industry Associations,
User Associations, Unions,
Insurance Companies, Courts

Regulations
Standards
Certification
Legal penalties
Case Law

Accident and incident reports
Operations reports
Maintenance Reports
Change reports
Whistleblowers

Company Management

Safety Policy
Standards
Resources

Operations Reports

Operations Management

Work Instructions

Change requests
Audit reports
Problem reports

Operating Assumptions
Operating Procedures

Operating Process

Human Controller(s)

Automated Controller

Actuator(s)

Sensor(s)

Physical Process

Revised operating procedures

Software revisions
Hardware replacements

Problem Reports
Incidents

(Leveson, 2012)

This is an engineered system too!
Need to identify and address the structural flaws!

Classification of Causal Factors in STAMP

Principles from Control Theory

Four conditions required to effect control over a system:

Goal Condition: The controller must have a goal or goals (e.g., to maintain a setpoint)

Action Condition: The controller must be able to affect the system state

Observability Condition: The controller must be able to ascertain the state of the system.

Model Condition: The controller must have (or contain) a model of the system

These conditions must be met for
effective management of safety (and security!)

Ashby, 1957

Principles from Control Theory

Four conditions required to effect control over a system:

Goal Condition: The controller must have a goal or goals (e.g., to maintain a setpoint)

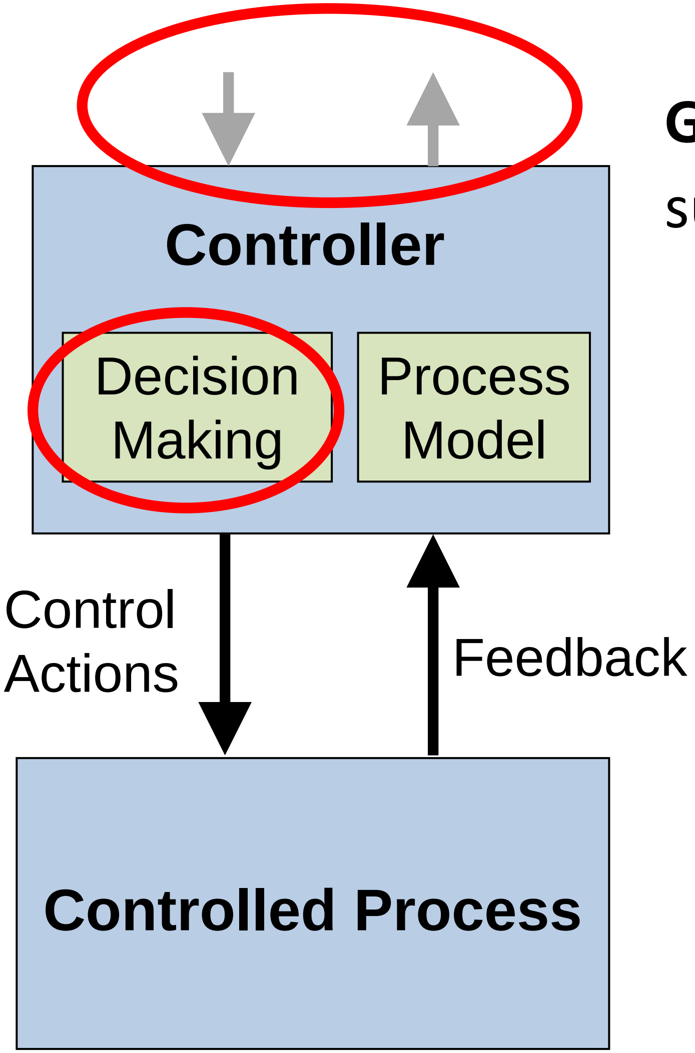
Action Condition: The controller must be able to affect the system state

Observability Condition: The controller must be able to ascertain the state of the system.

Model Condition: The controller must have (or contain) a model of the system

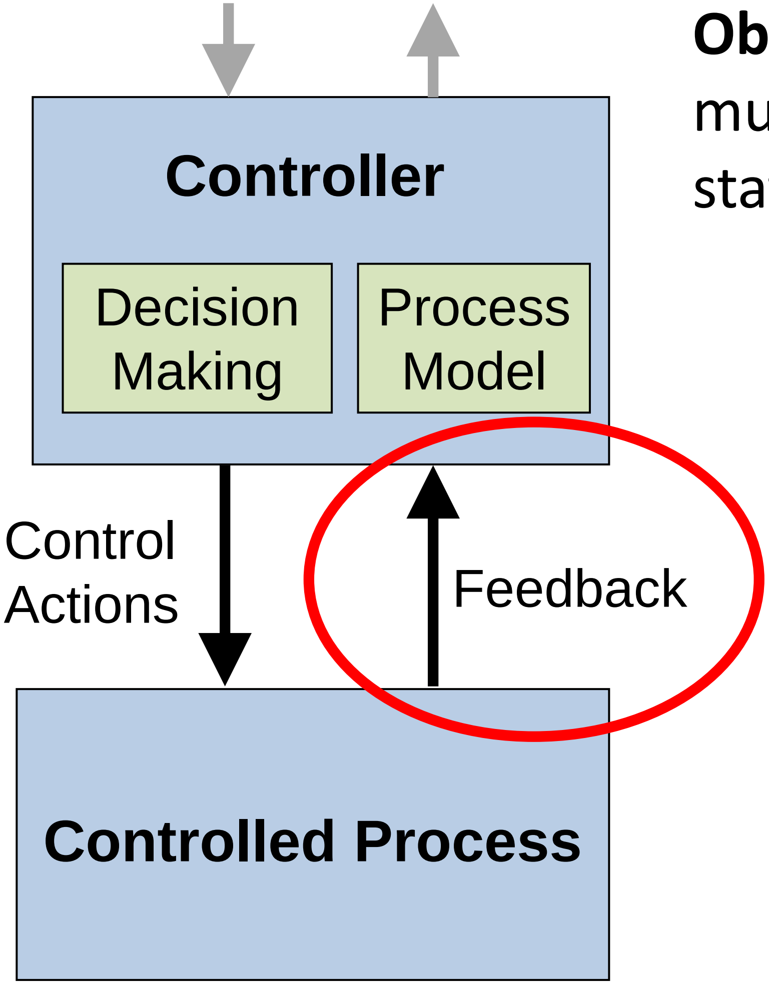
Potential targets for an adversary!

Ashby, 1957



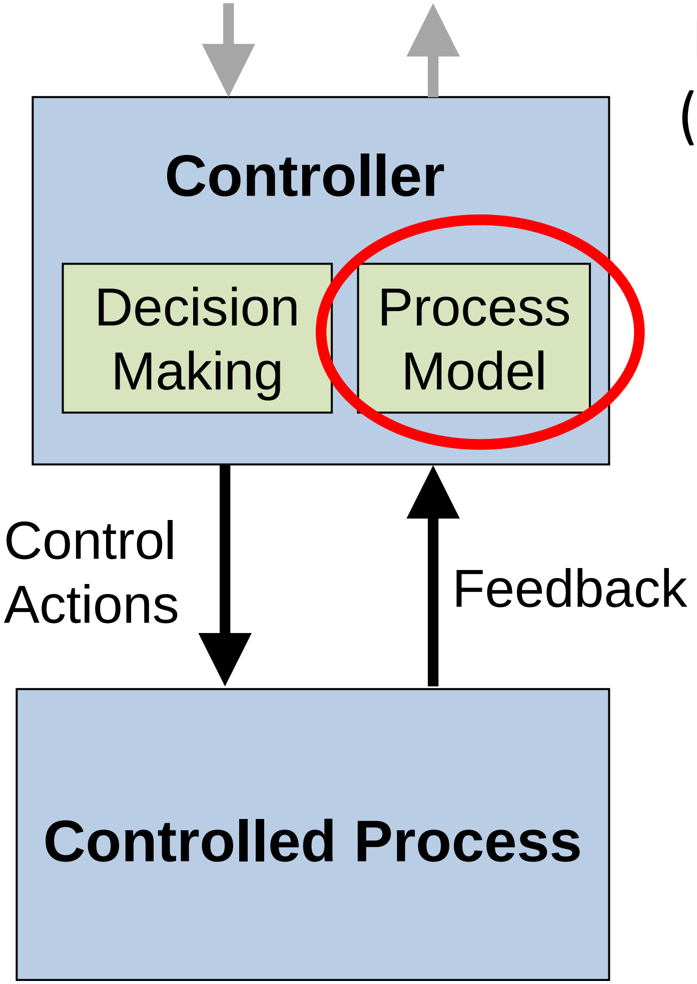
Goal Condition: The controller must have suitable goals (e.g., to maintain a setpoint)

(Leveson, 2012)



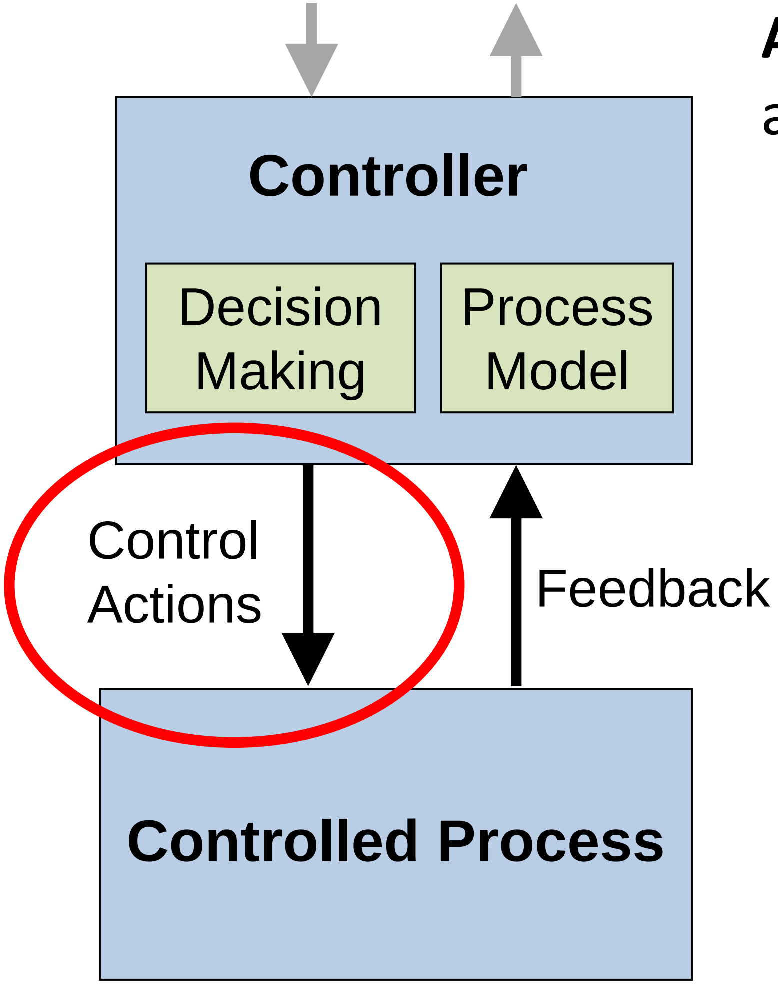
Observability Condition: The controller must be able to ascertain the relevant states of the system.

(Leveson, 2012)



Model Condition: The controller must have (or contain) a model of the system

(Leveson, 2012)

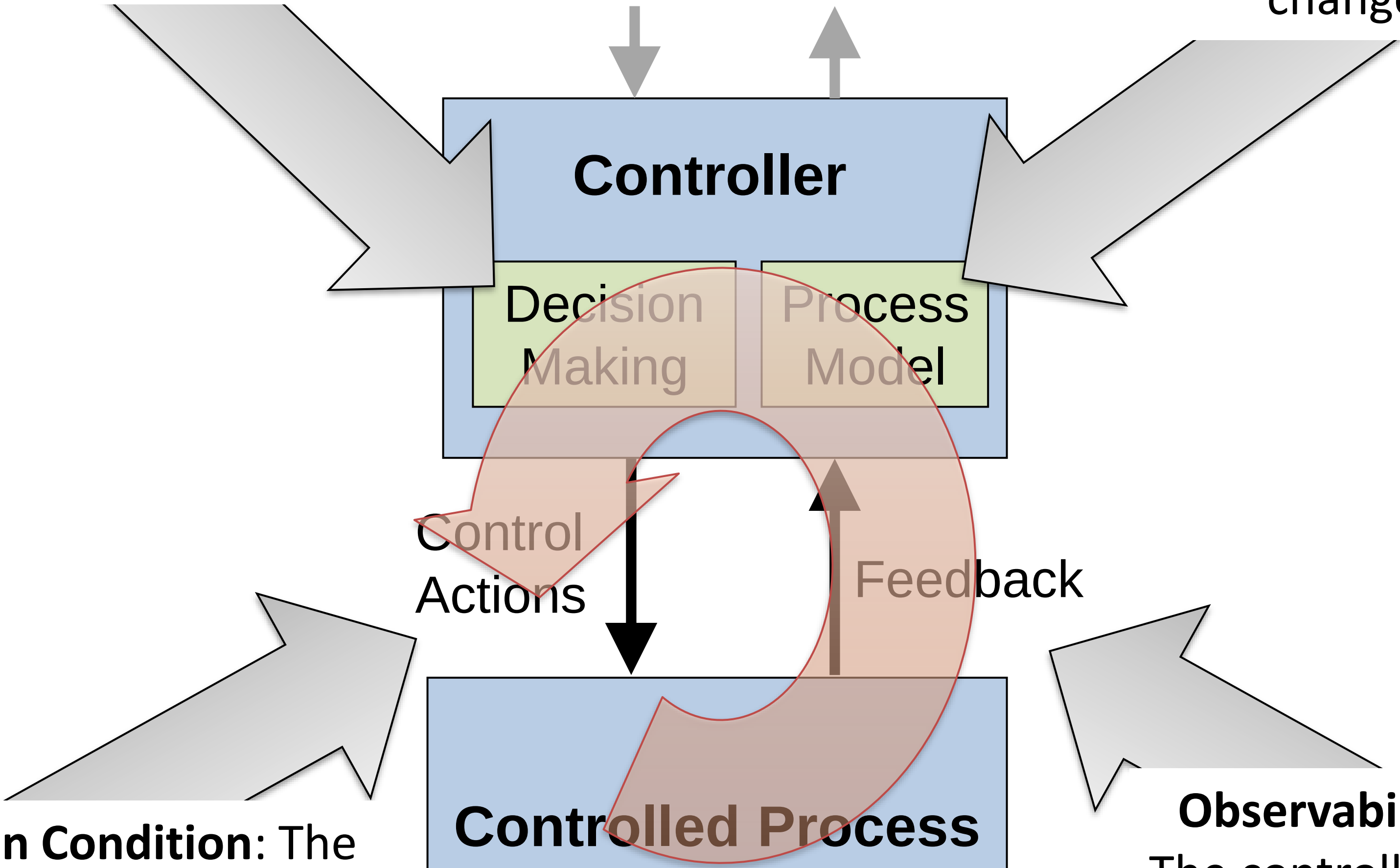


Action Condition: The controller must be able to affect the system state

(Leveson, 2012)

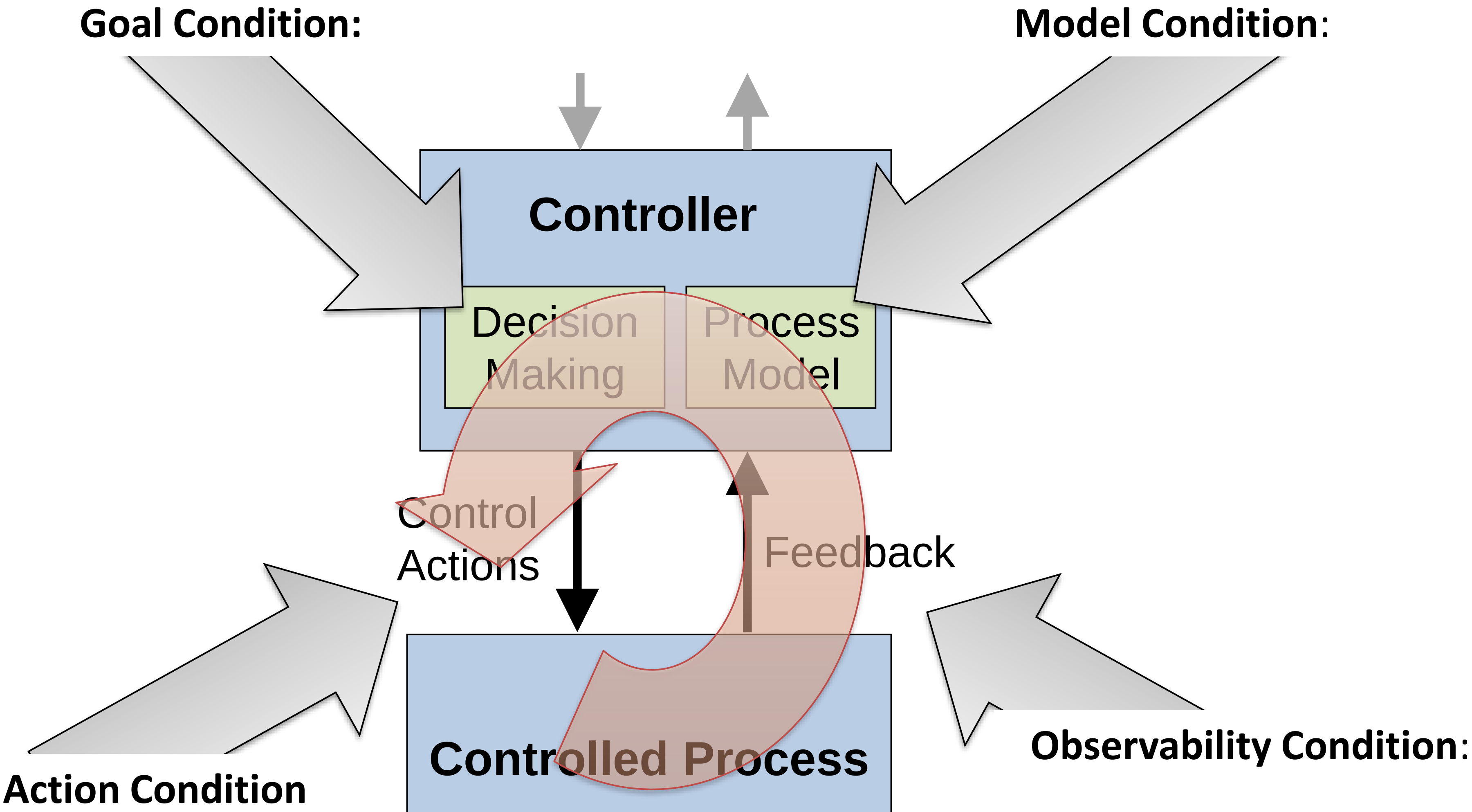
Goal Condition: The controller must have and prioritize appropriate goals

Model Condition: The controller must have a model of the system state and how it can change

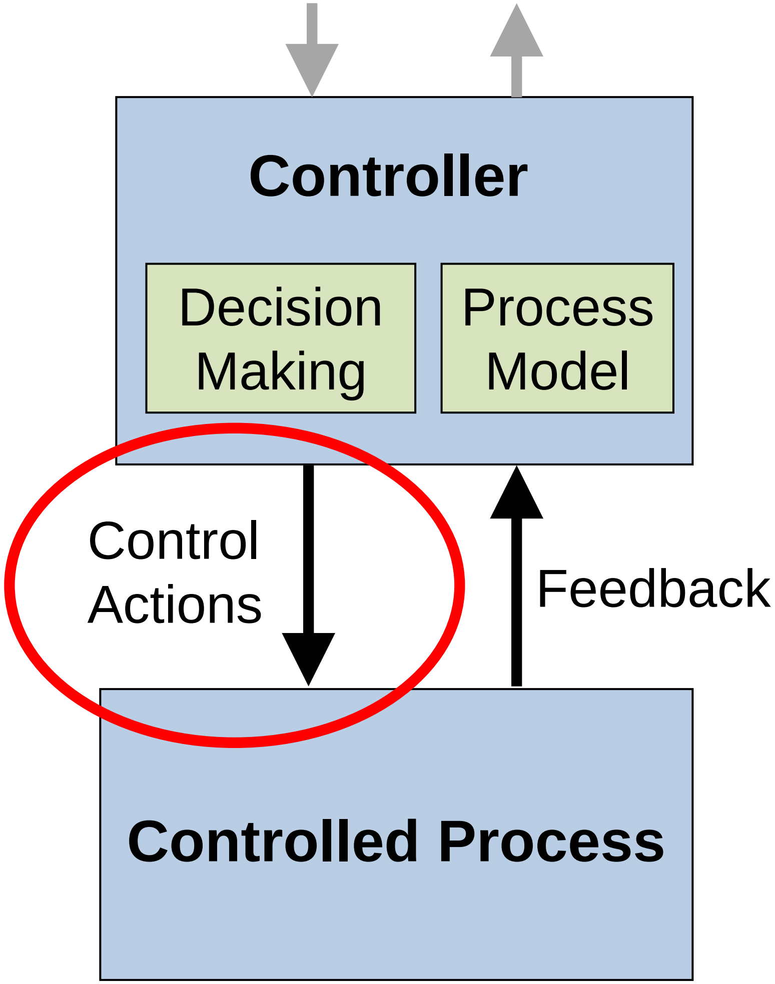


Action Condition: The controller must have appropriate authority and control to achieve the goals

Observability Condition: The controller must be able to ascertain the state of the system.



Accidents & incidents events occur because these conditions were broken!



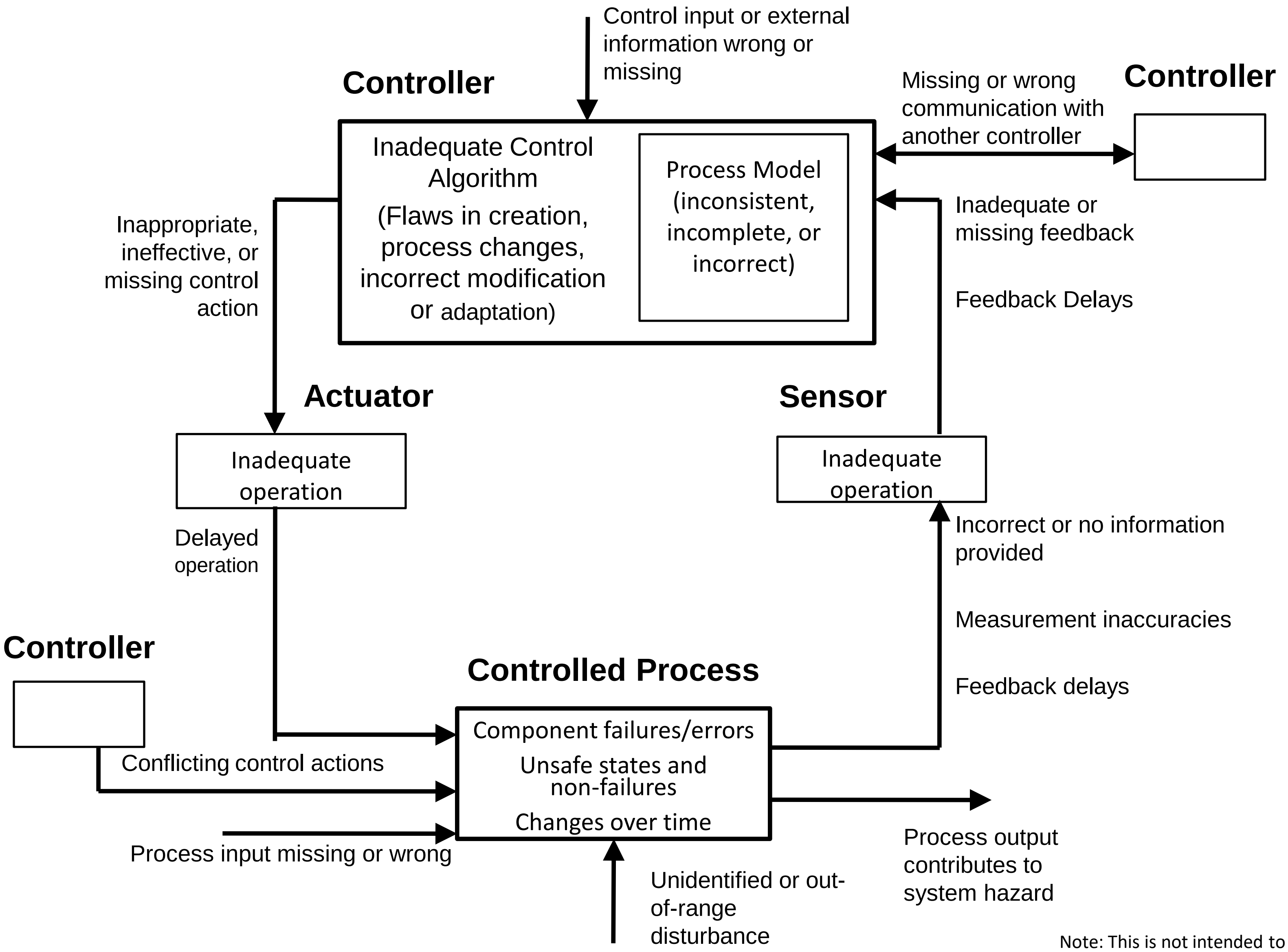
Unsafe Control Actions (UCAs)

Control Actions may be Unsafe in 4 ways:

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

(Leveson, 2012)

Some Factors in Causal Scenarios

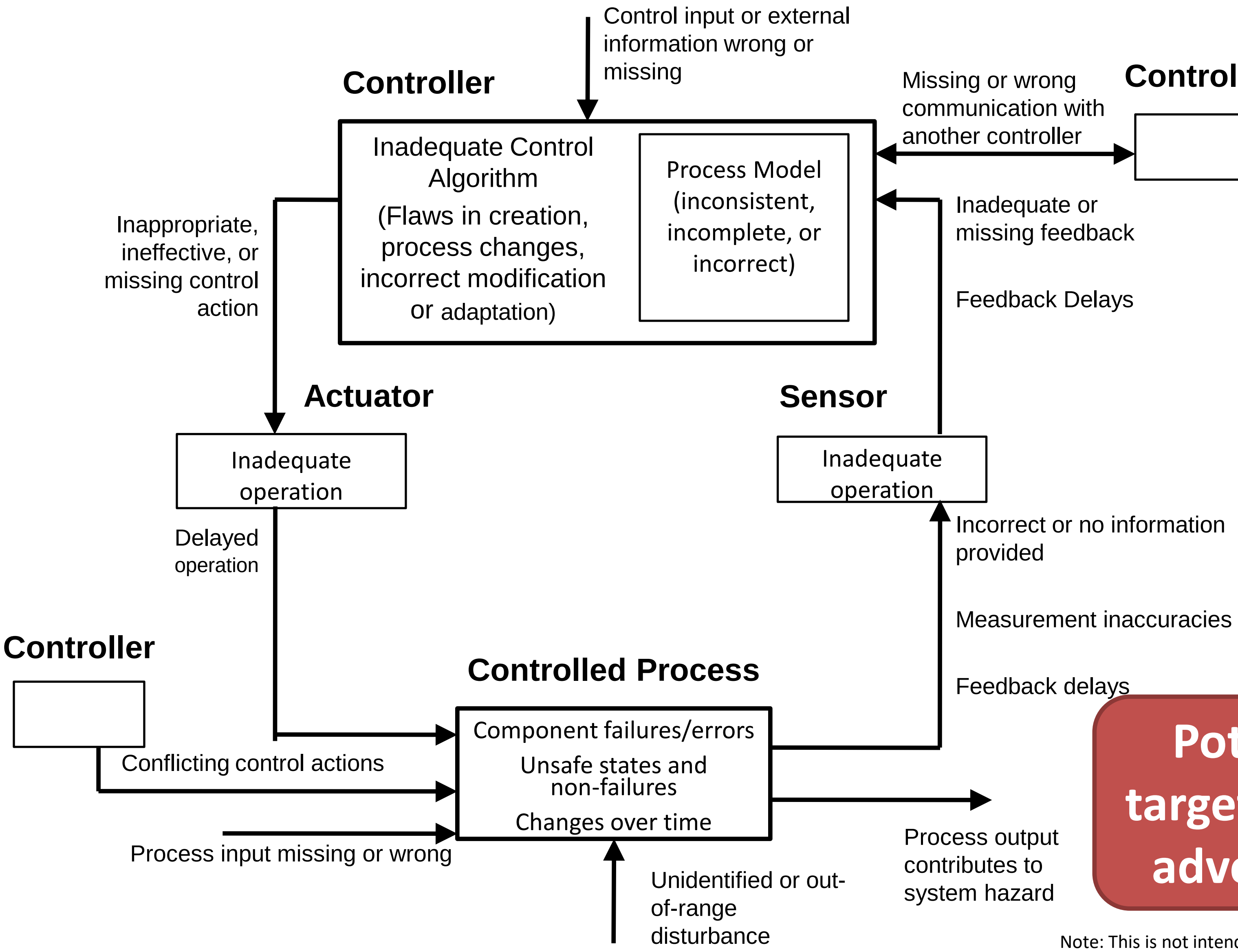


Control input or external information wrong or missing
Controller
Missing or wrong communication with another controller
Controller
Inadequate Control Algorithm
(Flaws in creation, process changes, incorrect modification or adaptation)
Process Model (inconsistent, incomplete, or incorrect)
Inappropriate,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

(Leveson, 2012)

Note: This is not intended to be complete, but it provides a starting point. You will need to tailor the specific factors relevant to your application.

Some Factors in Causal Scenarios



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

Inadequate or missing feedback

Feedback Delays

Inappropriate, ineffective, or missing control action

Actuator

Inadequate operation

Delayed operation

Sensor

Inadequate operation

Incorrect or no information provided

Measurement inaccuracies

Feedback delays

Controller

Controlled Process

Component failures/errors

Unsafe states and non-failures

Changes over time

Conflicting control actions

Process input missing or wrong

Unidentified or out-of-range disturbance

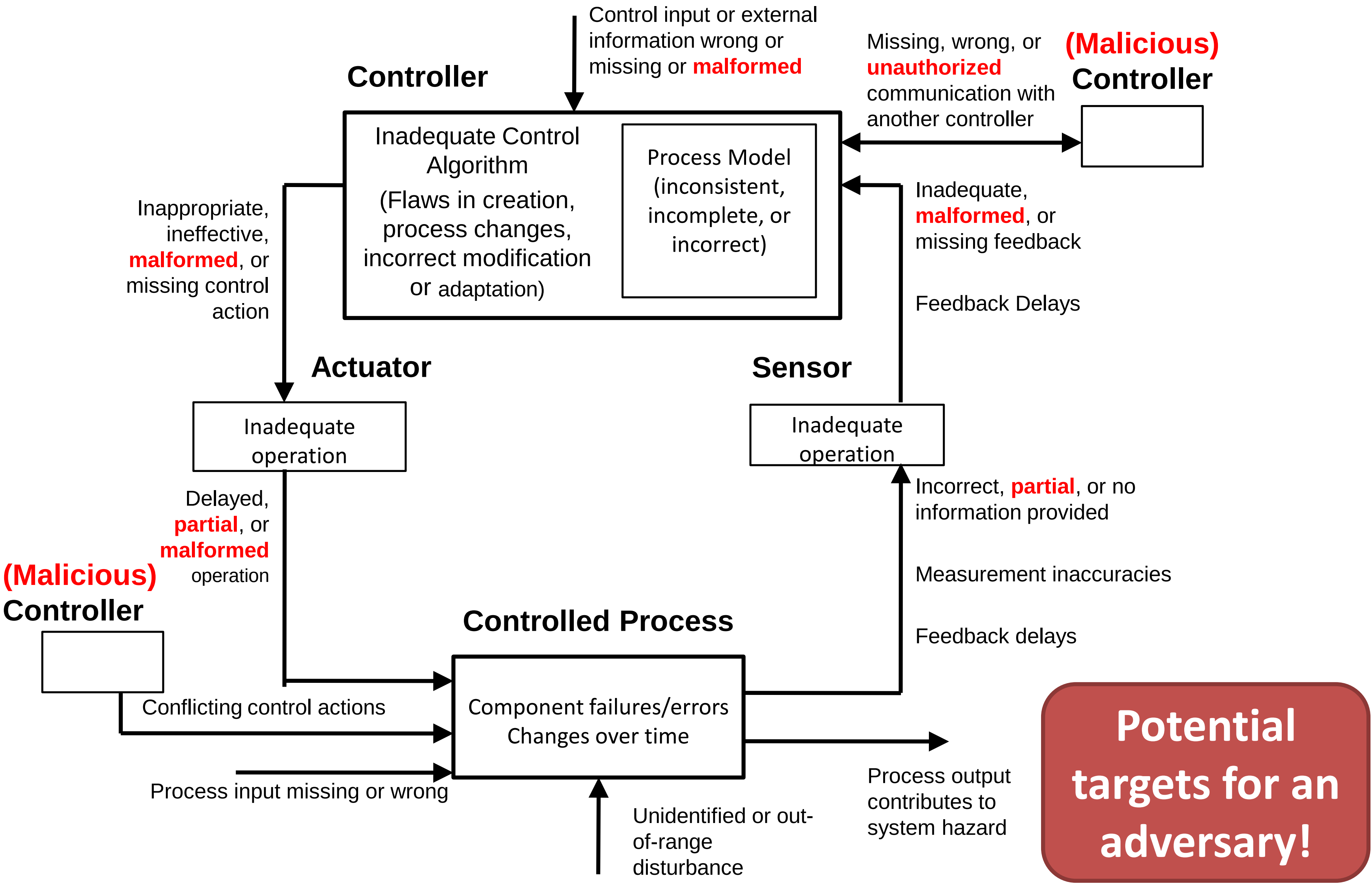
Process output contributes to system hazard

Potential targets for an adversary?

Note: This is not intended to be complete, but it provides a starting point. You will need to tailor the specific factors relevant to your application.

(Leveson, 2012)

Some Factors in Causal Scenarios



Control input or external information wrong or missing or **malformed**

Controller

Inadequate Control Algorithm (Flaws in creation, process changes, incorrect modification or adaptation)

Process Model (inconsistent, incomplete, or incorrect)

Missing, wrong, or **unauthorized** communication with another controller

(Malicious) Controller

Inappropriate, ineffective, **malformed**, or missing control action

Inadequate, **malformed**, or missing feedback

Feedback Delays

Actuator

Inadequate operation

Sensor

Inadequate operation

Delayed, **partial**, or **malformed** operation

Incorrect, **partial**, or no information provided

Measurement inaccuracies

Feedback delays

(Malicious) Controller

Controlled Process

Component failures/errors
Changes over time

Conflicting control actions

Process input missing or wrong

Unidentified or out-of-range disturbance

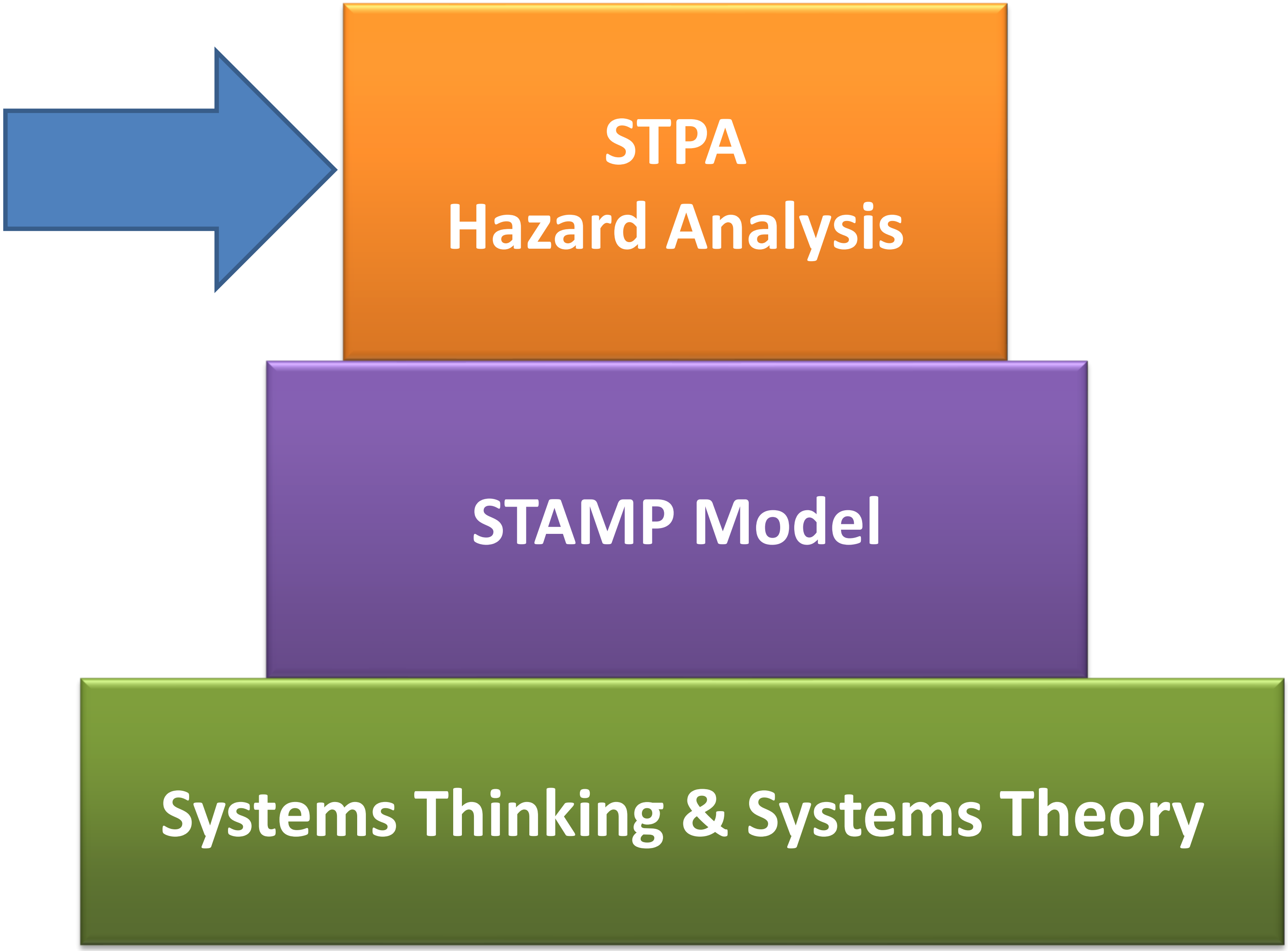
Process output contributes to system hazard

Potential targets for an adversary!

(Young, 2014)

- What is an accident model?
- What is STAMP?
- What is STPA?

System Theory, STAMP, STPA



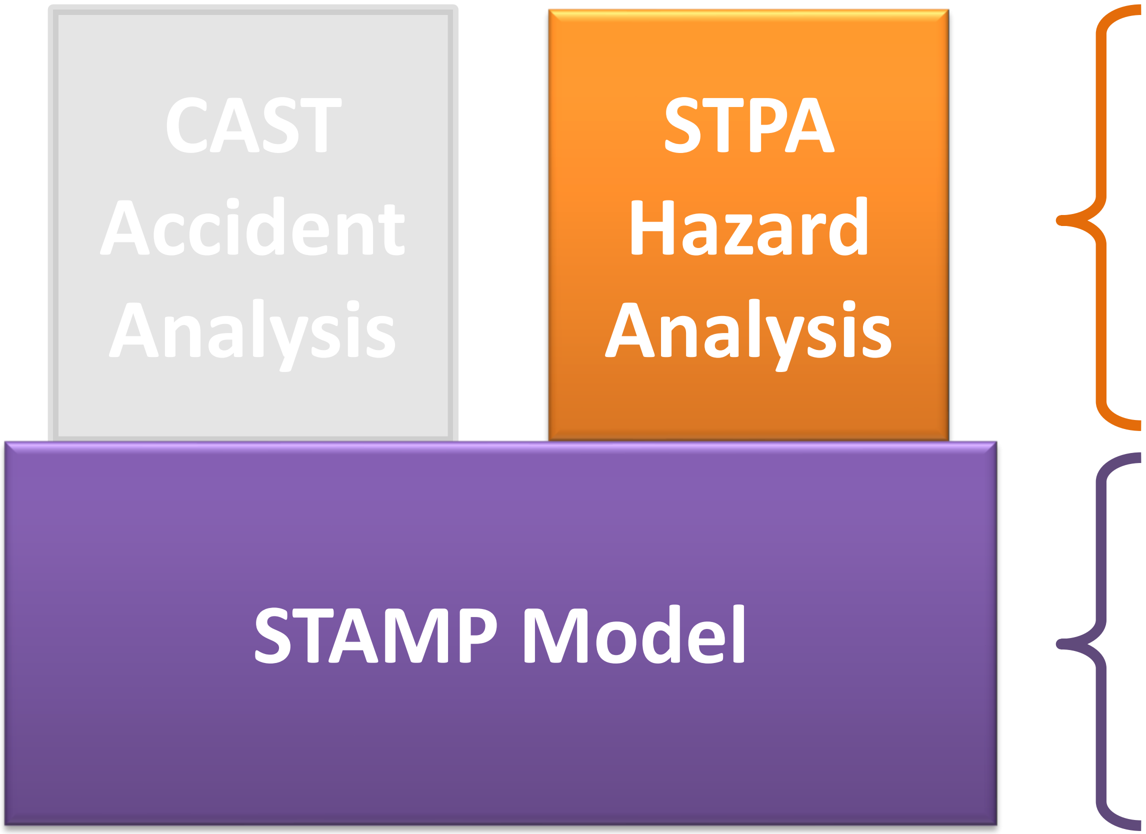
STPA
Hazard Analysis

STAMP Model

Systems Thinking & Systems Theory

(Leveson, 2012)

STAMP, STPA, and CAST



CAST
Accident
Analysis

STPA
Hazard
Analysis

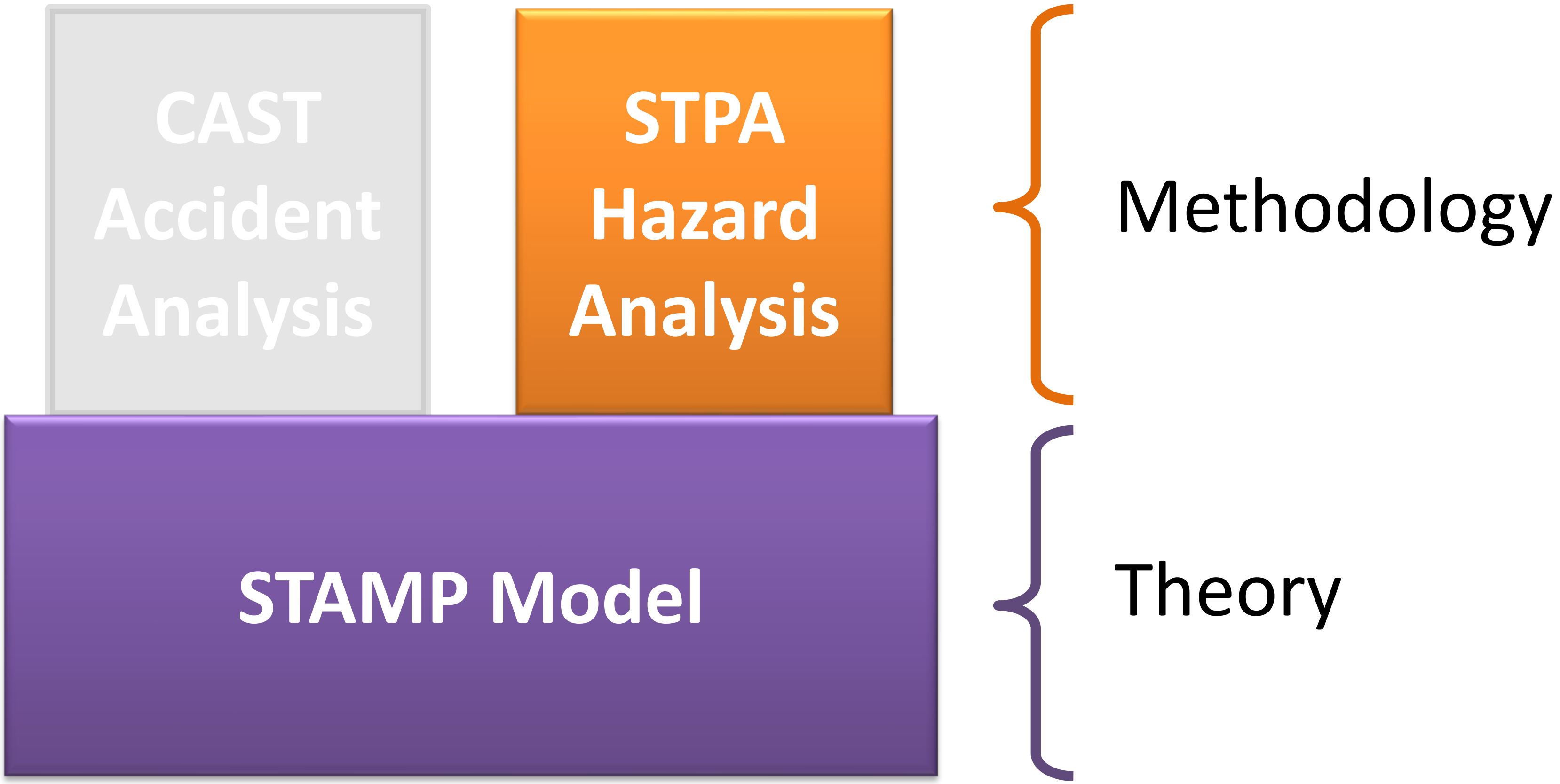
STAMP Model

How do we find inadequate control in a design?

Losses are caused by inadequate control

(Leveson, 2012)

STAMP, STPA, and CAST



CAST Accident Analysis

STPA Hazard Analysis

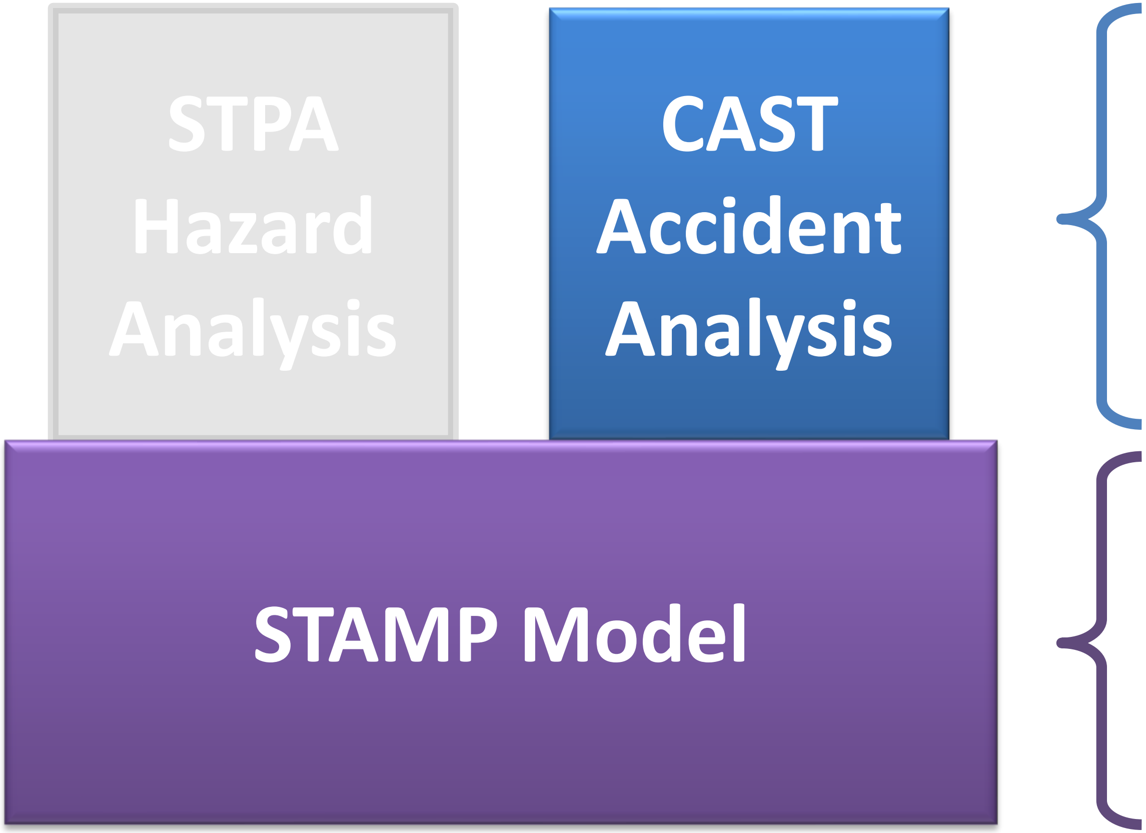
STAMP Model

Methodology

Theory

(Leveson, 2012)

STAMP, STPA, and CAST



STPA Hazard Analysis

CAST Accident Analysis

STAMP Model

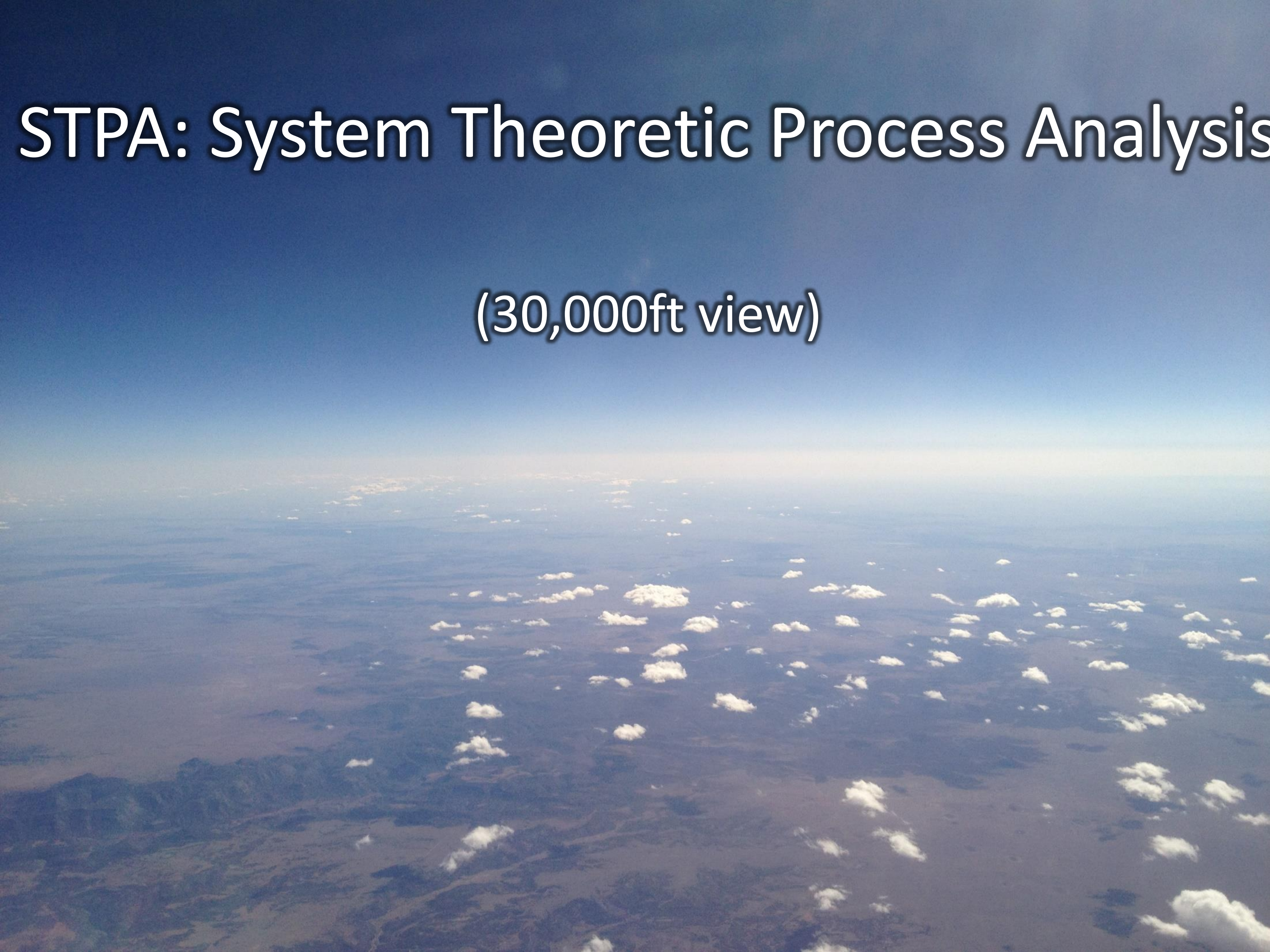
How do we find all causes of a past loss event?

Losses are caused by inadequate control

(Leveson, 2012)

STPA: System Theoretic Process Analysis

(30,000ft view)



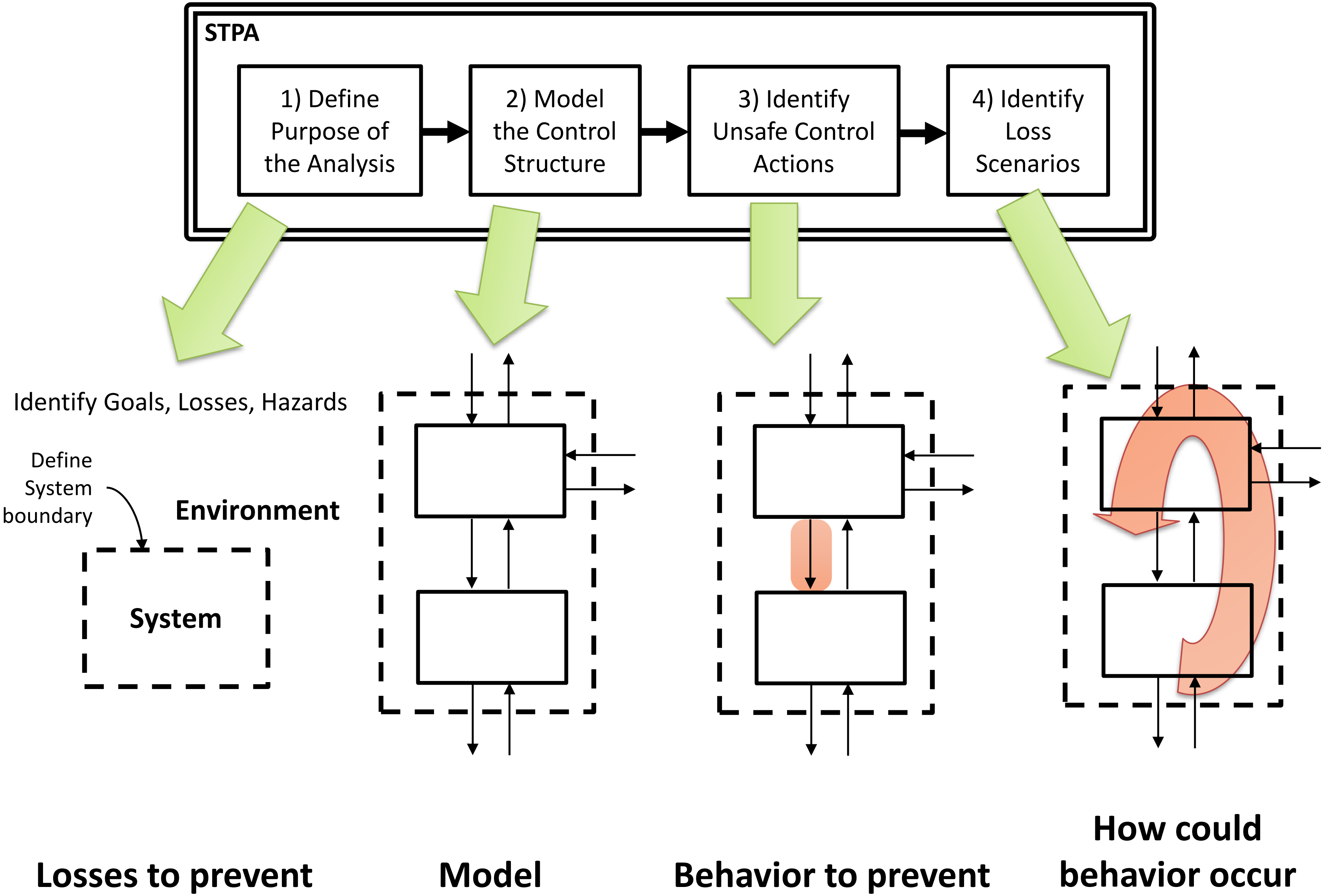
System-Theoretic Process Analysis (STPA)

STPA is a technique for safety-driven development and assessment

STPA anticipates hazardous scenarios caused by:

- Unsafe decision-making
- Software, computers, and automation
- Human error/confusion
- Flawed assumptions
- Missing design requirements
- Interactions between systems
- Etc.

Thomas, 2013



STPA

1) Define Purpose of the Analysis

2) Model the Control Structure

3) Identify Unsafe Control Actions

4) Identify Loss Scenarios

Identify Goals, Losses, Hazards

Define System boundary

Environment

System

Losses to prevent

Model

Behavior to prevent

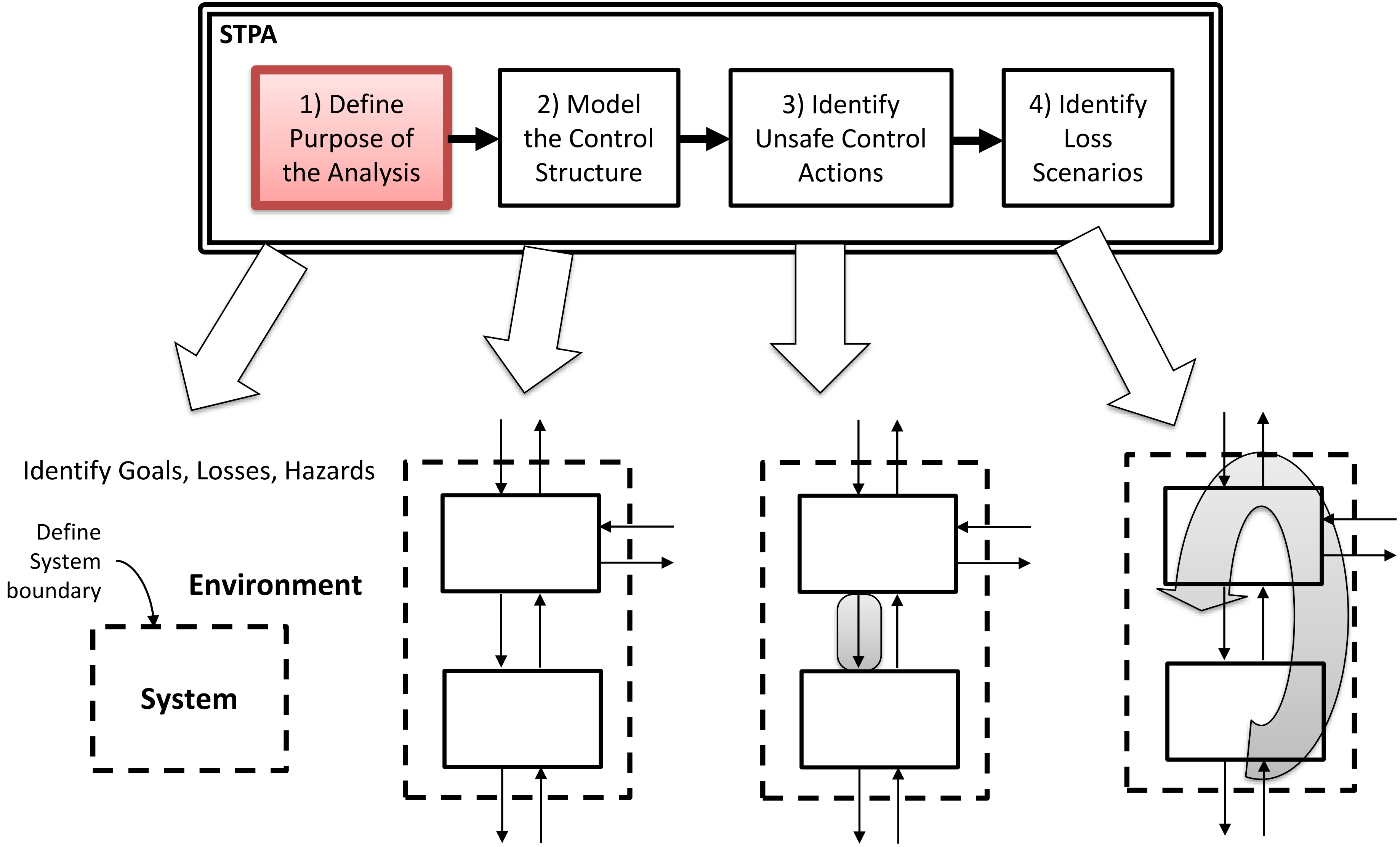
How could behavior occur

(Leveson and Thomas, 2018)

STPA: System Theoretic Process Analysis

(10,000ft view)

STPA



STPA

1) Define Purpose of the Analysis

2) Model the Control Structure

3) Identify Unsafe Control Actions

4) Identify Loss Scenarios

Identify Goals, Losses, Hazards

Define System boundary

Environment

System

(Leveson and Thomas, 2018)

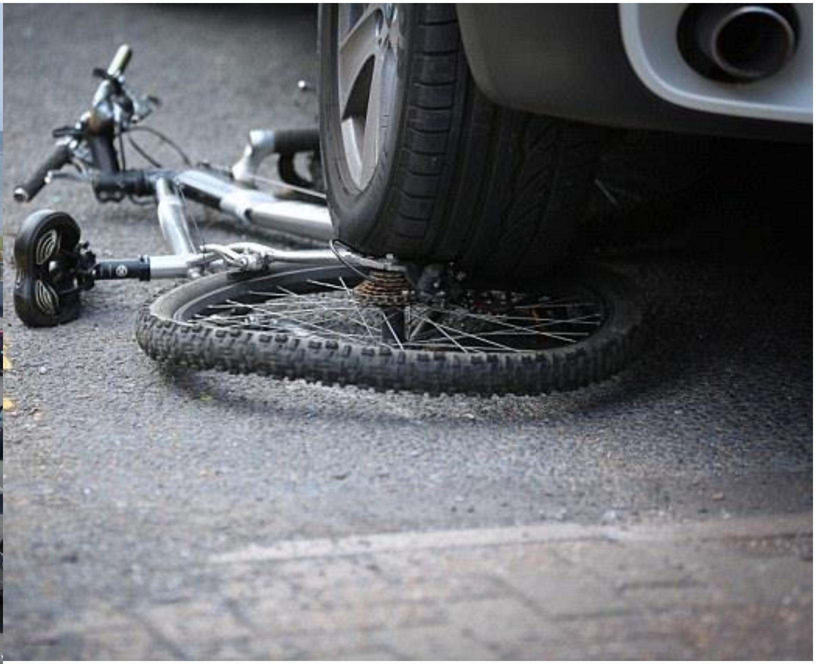
Automotive Example

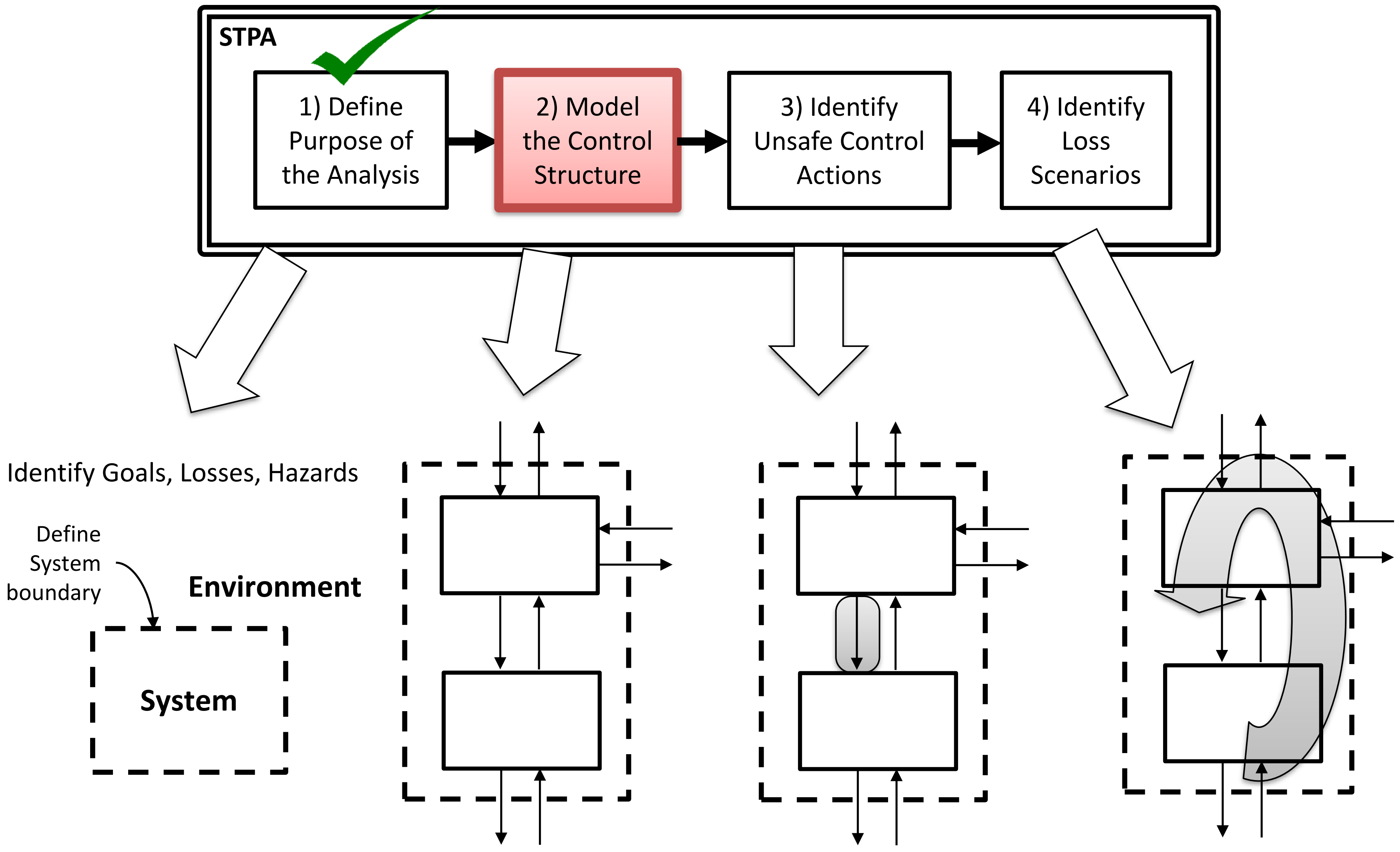
- Stakeholder Losses to prevent
  - L-1. Loss of life or serious injury to people
  - L-2. Damage to the vehicle or objects outside the vehicle

A Stakeholder Loss ("Loss") involves something of value to stakeholders. It is a loss that is unacceptable to stakeholders.

Automotive Example

- Stakeholder Losses
  - L-1. Loss of life or serious injury to people
  - L-2. Damage to the vehicle or objects outside the vehicle
  - L-3: Loss of mission (transportation)
  - L-4: Loss of customer satisfaction
  - Etc.





STPA

1) Define Purpose of the Analysis

2) Model the Control Structure

3) Identify Unsafe Control Actions

4) Identify Loss Scenarios

Identify Goals, Losses, Hazards

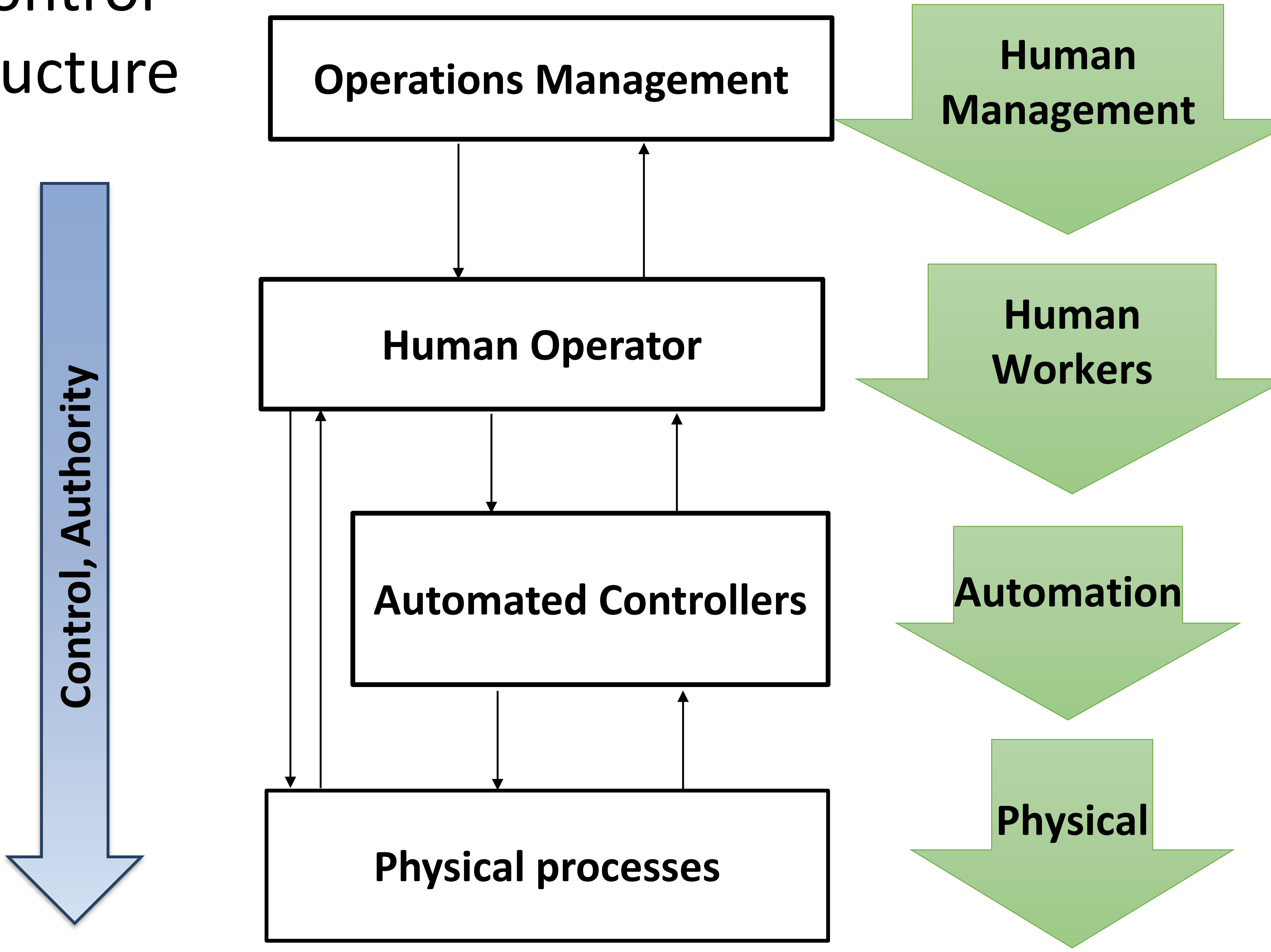
Define System boundary

Environment

System

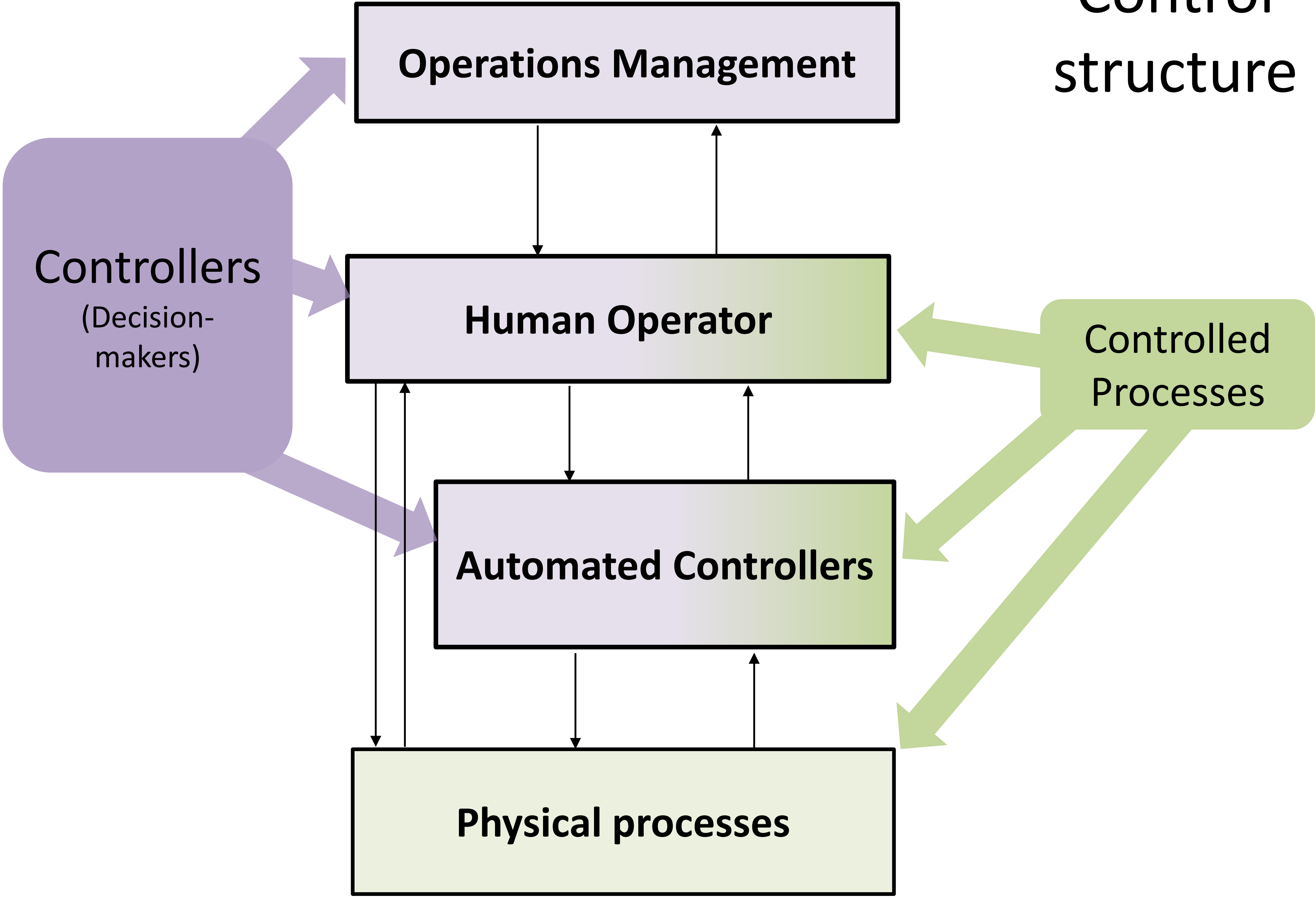
(Leveson and Thomas, 2018)

Control structure



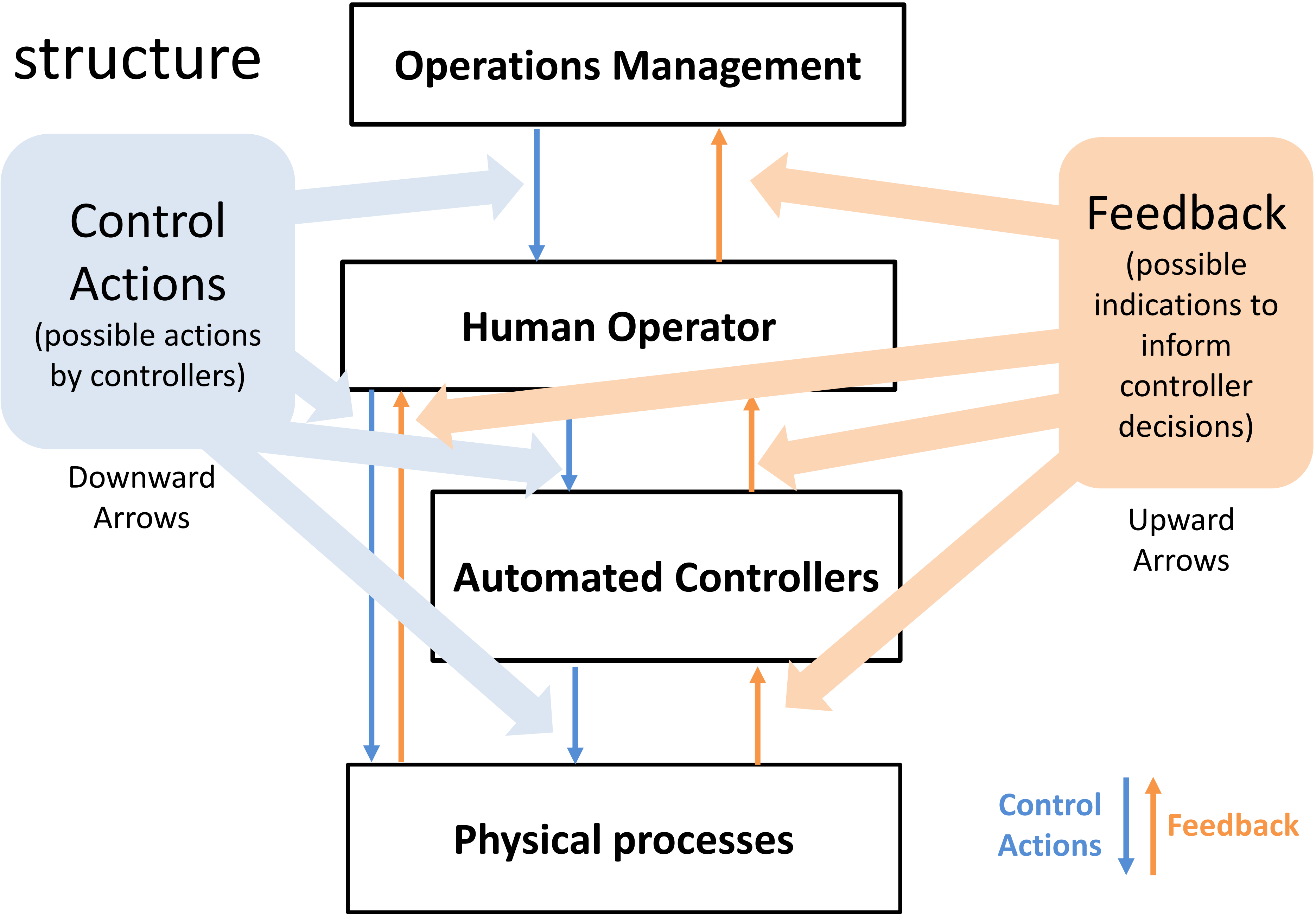
John Thomas, 2019

Control structure



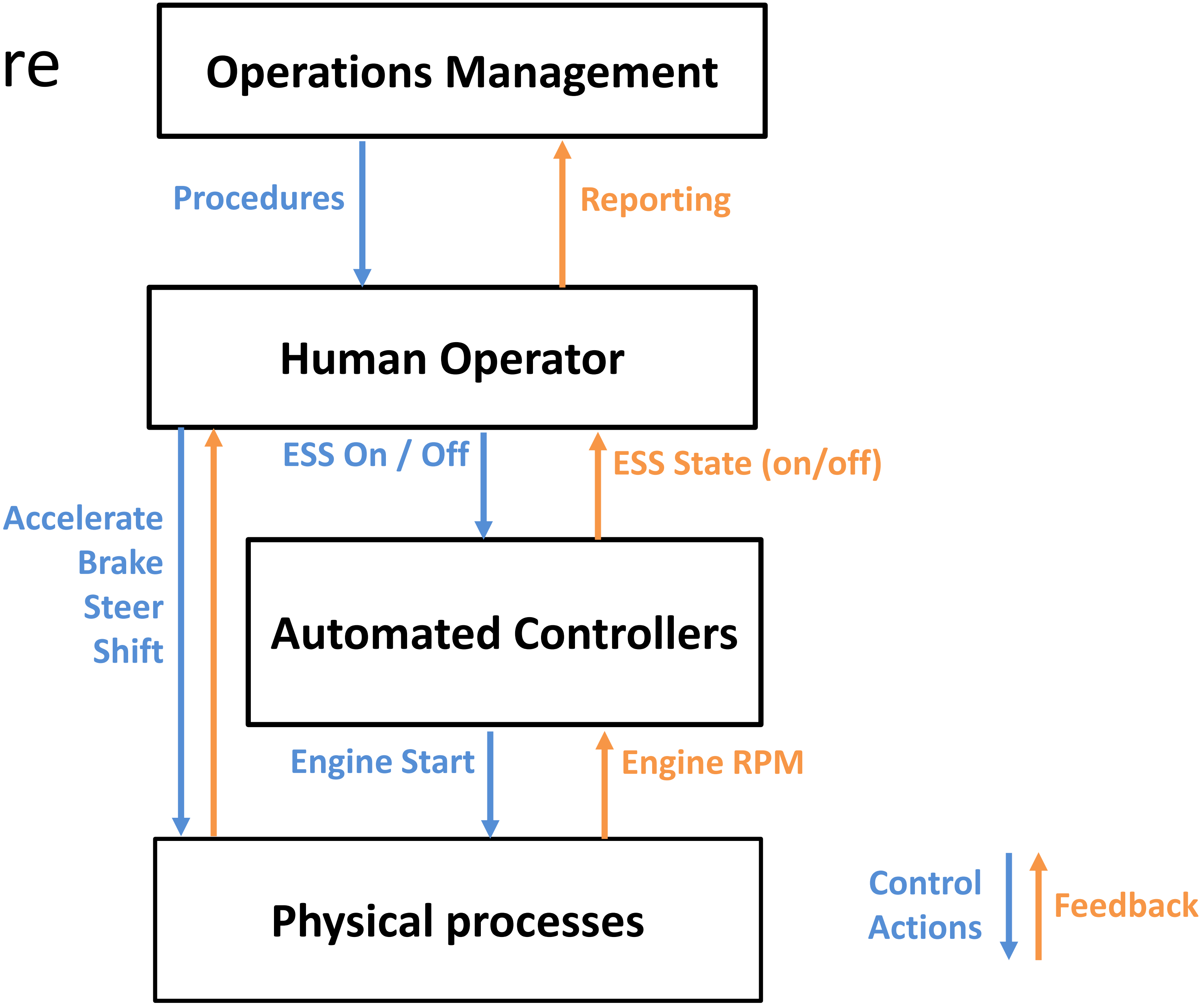
John Thomas, 2019

Control structure



John Thomas, 2019

Control structure



Operations Management

Procedures

Reporting

Human Operator

ESS On / Off

ESS State (on/off)

Accelerate
Brake
Steer
Shift

Automated Controllers

Engine Start

Engine RPM

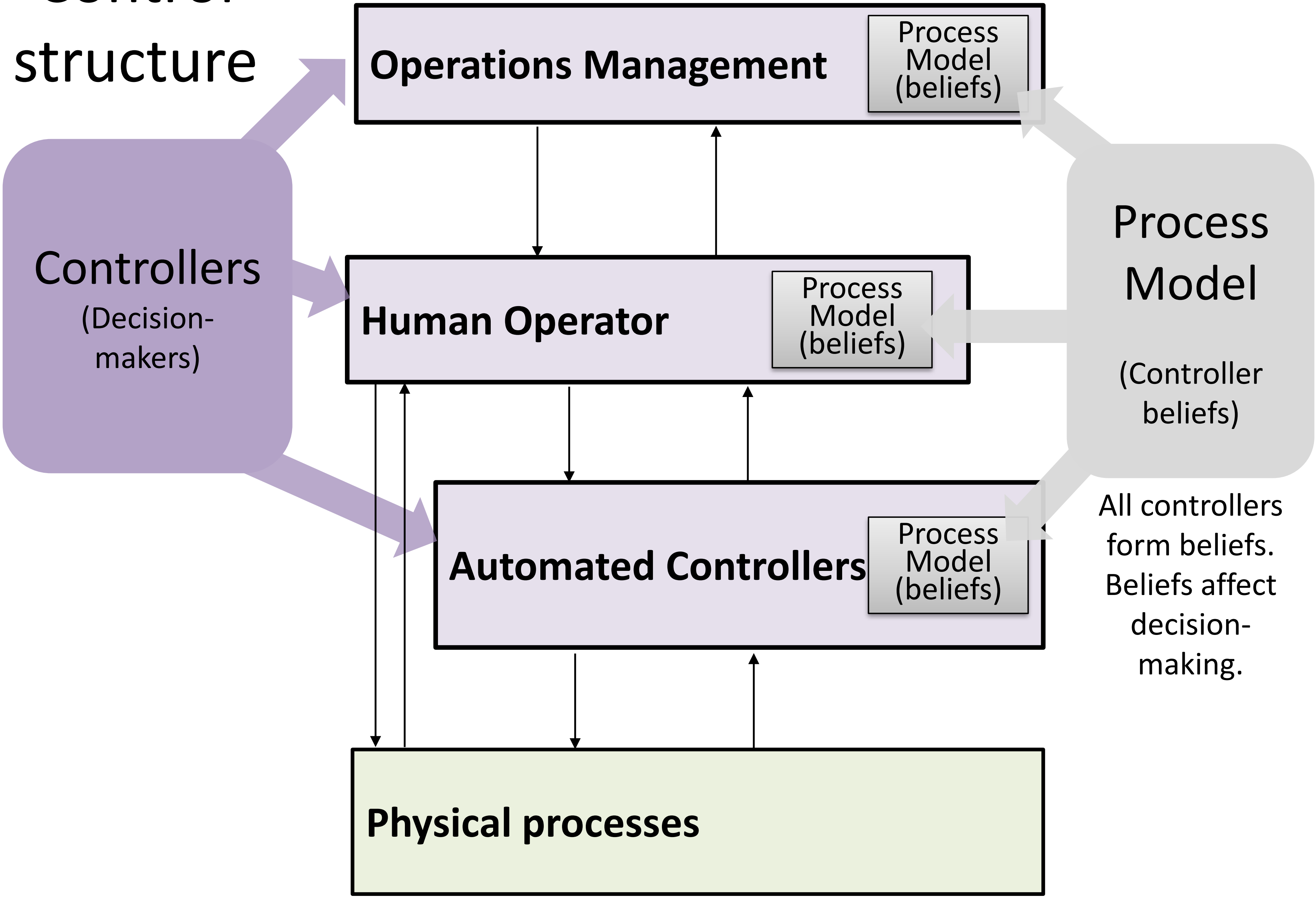
Physical processes

Control Actions

Feedback

John Thomas, 2019

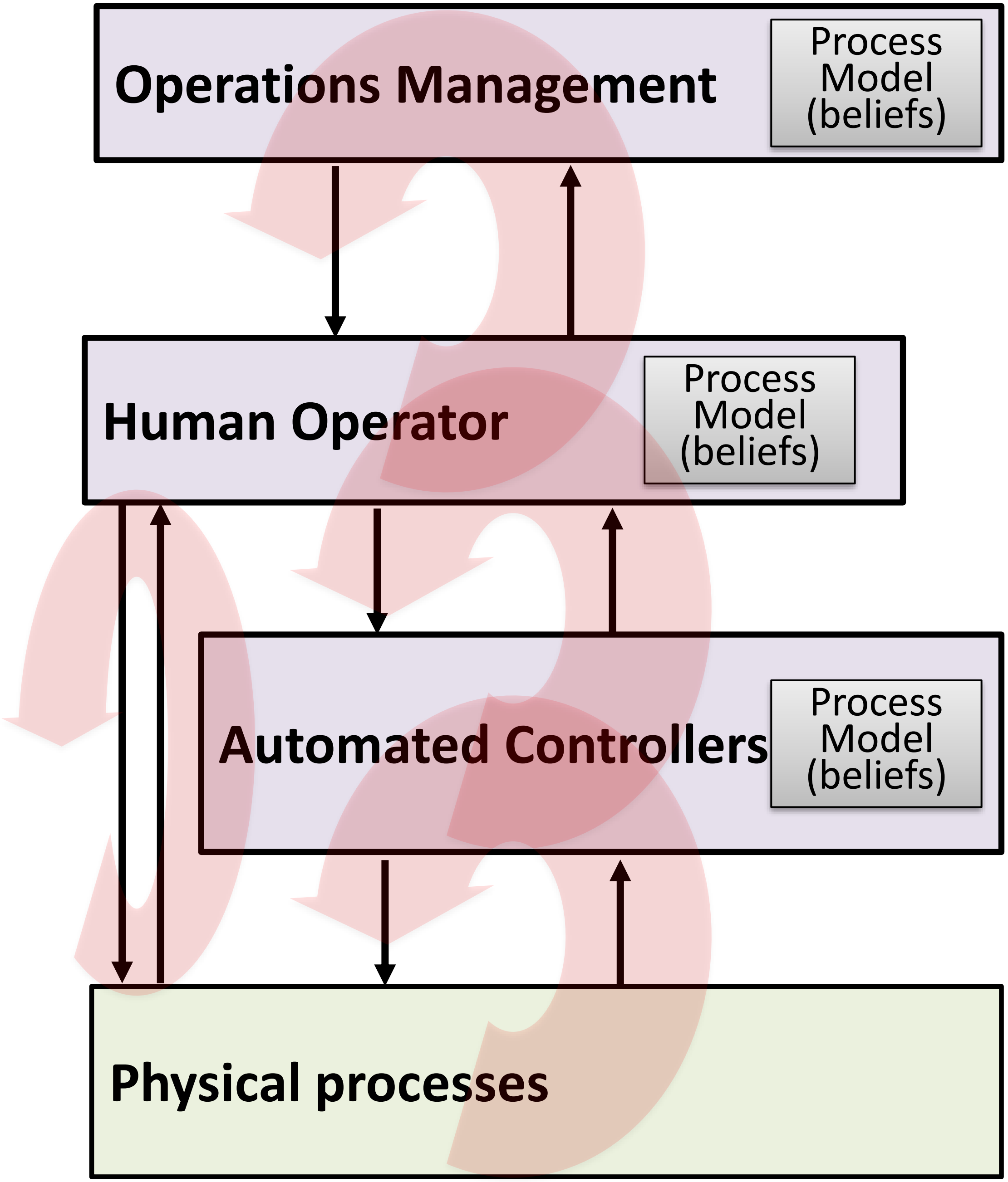
Control structure



All controllers form beliefs. Beliefs affect decision-making.

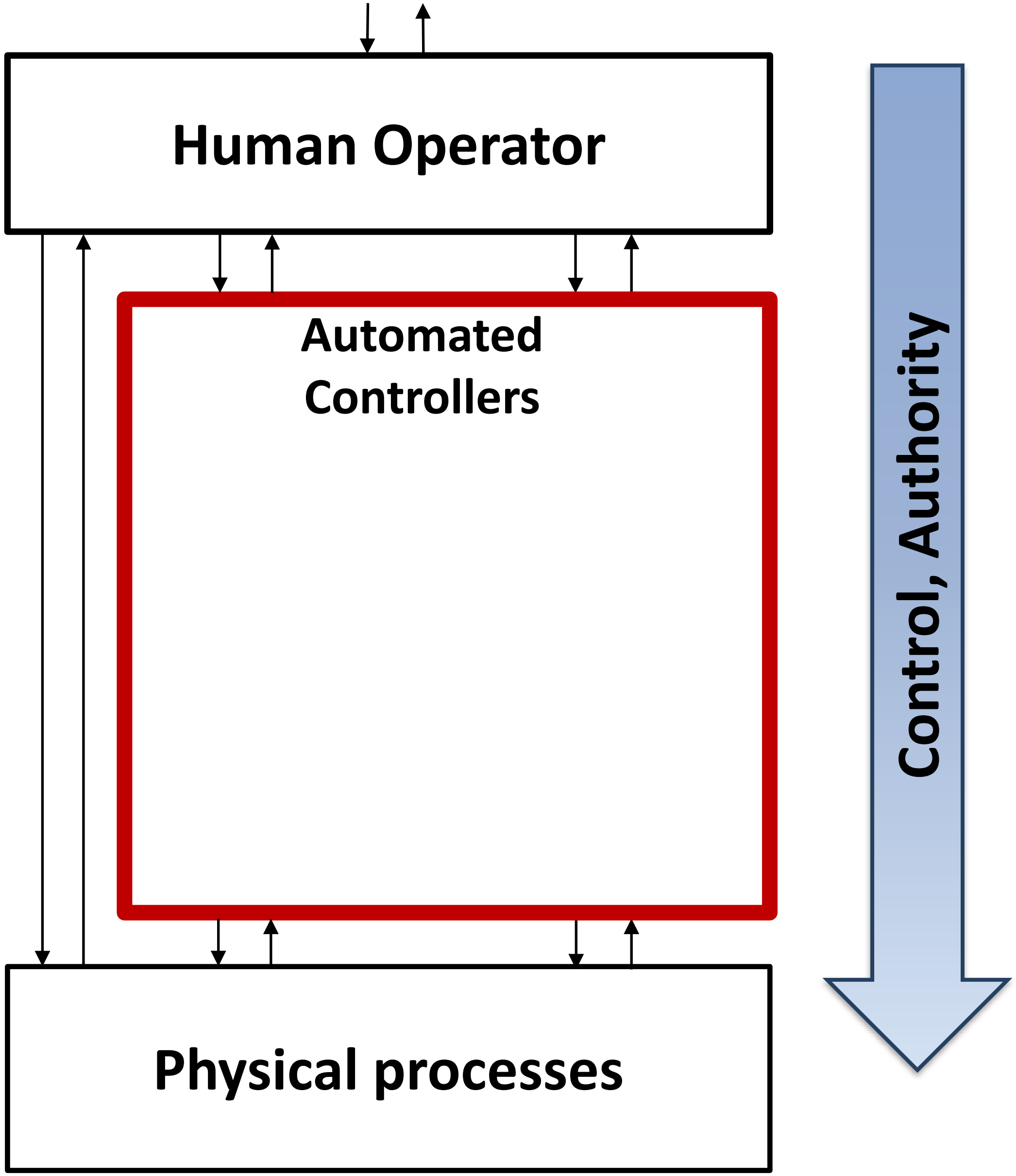
John Thomas, 2019

Control structure

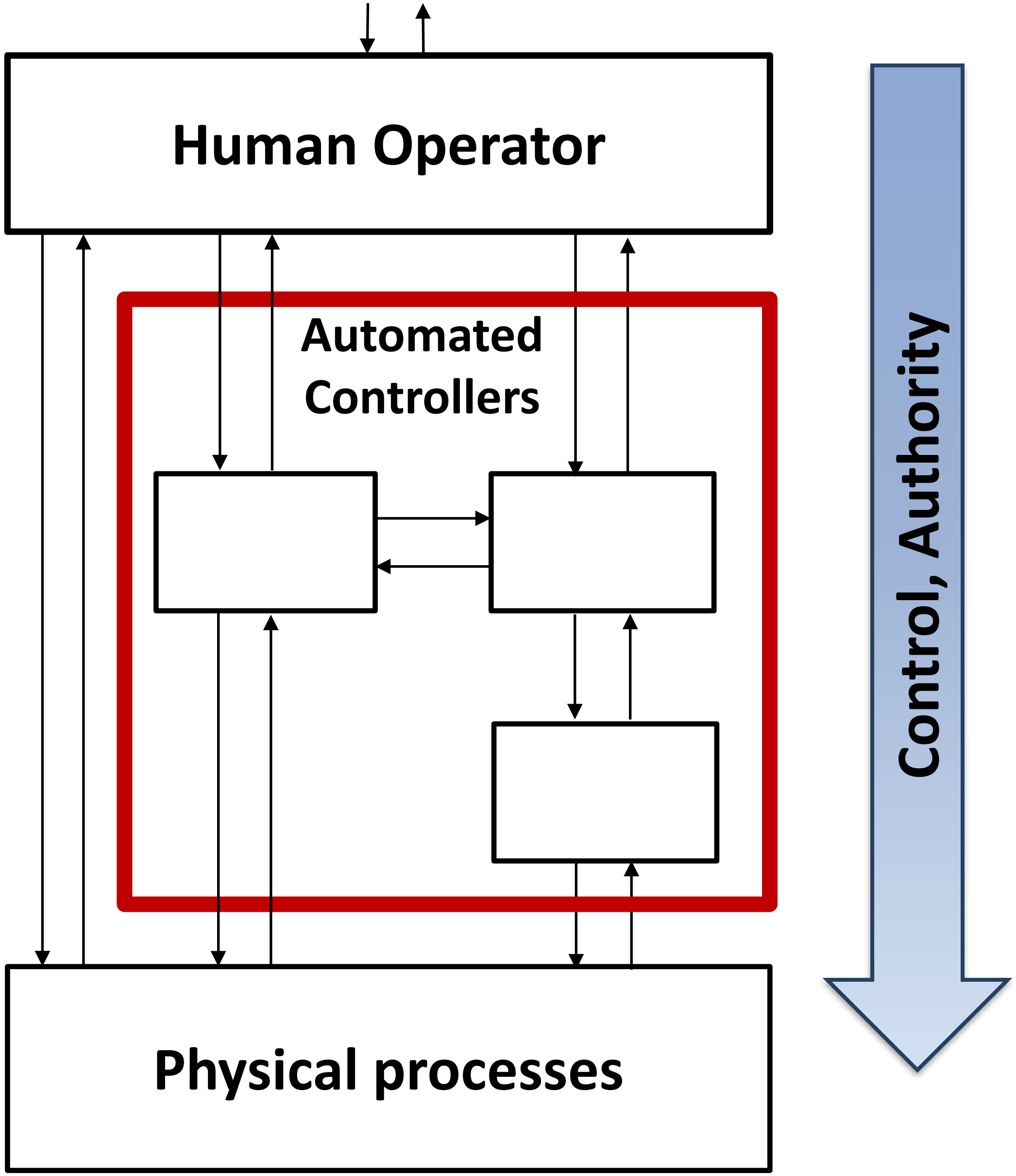


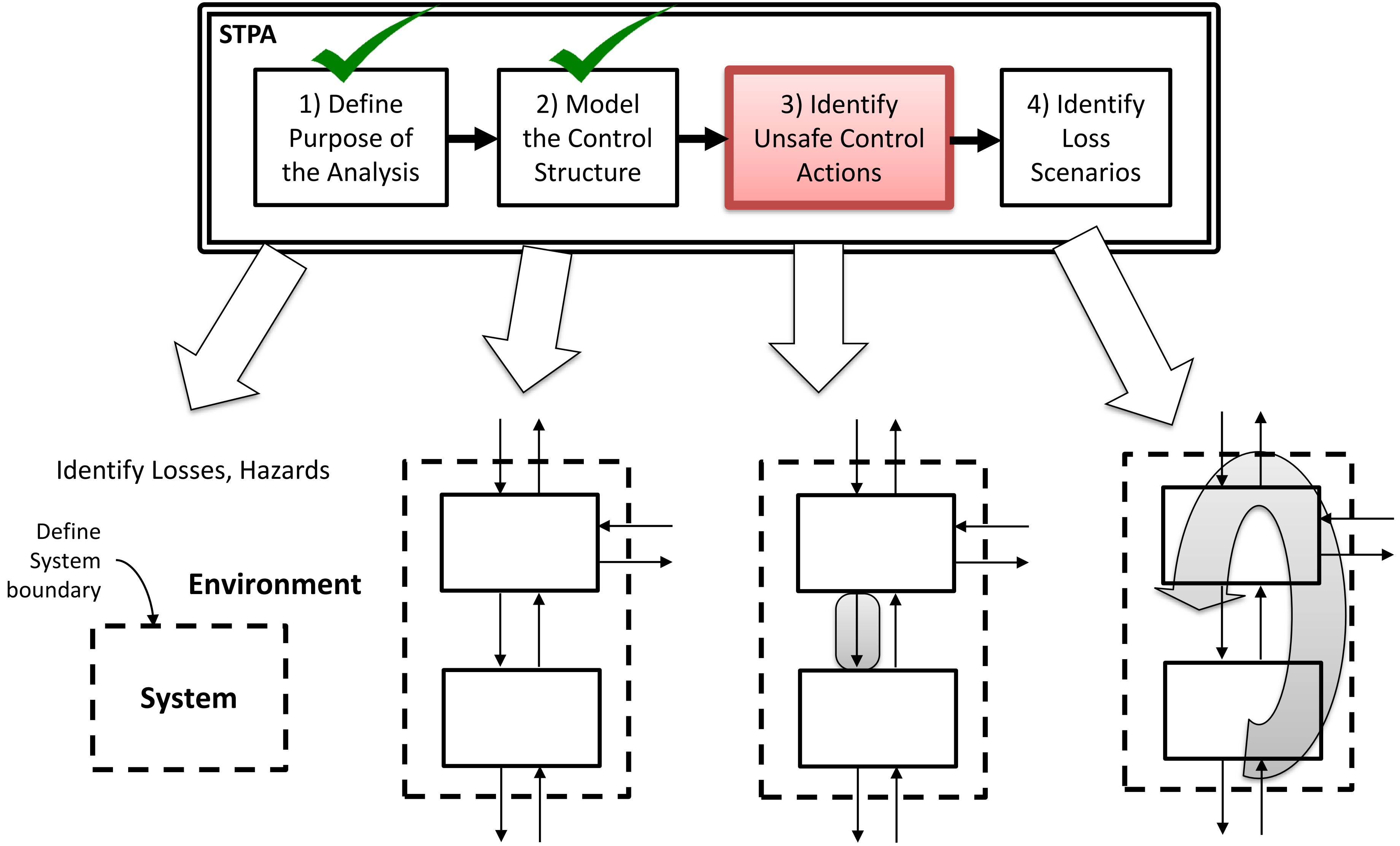
John Thomas, 2019

“Zooming in” to create more detailed control structure



“Zooming in” to create more detailed control structure





STPA

1) Define Purpose of the Analysis

2) Model the Control Structure

3) Identify Unsafe Control Actions

4) Identify Loss Scenarios

Identify Losses, Hazards

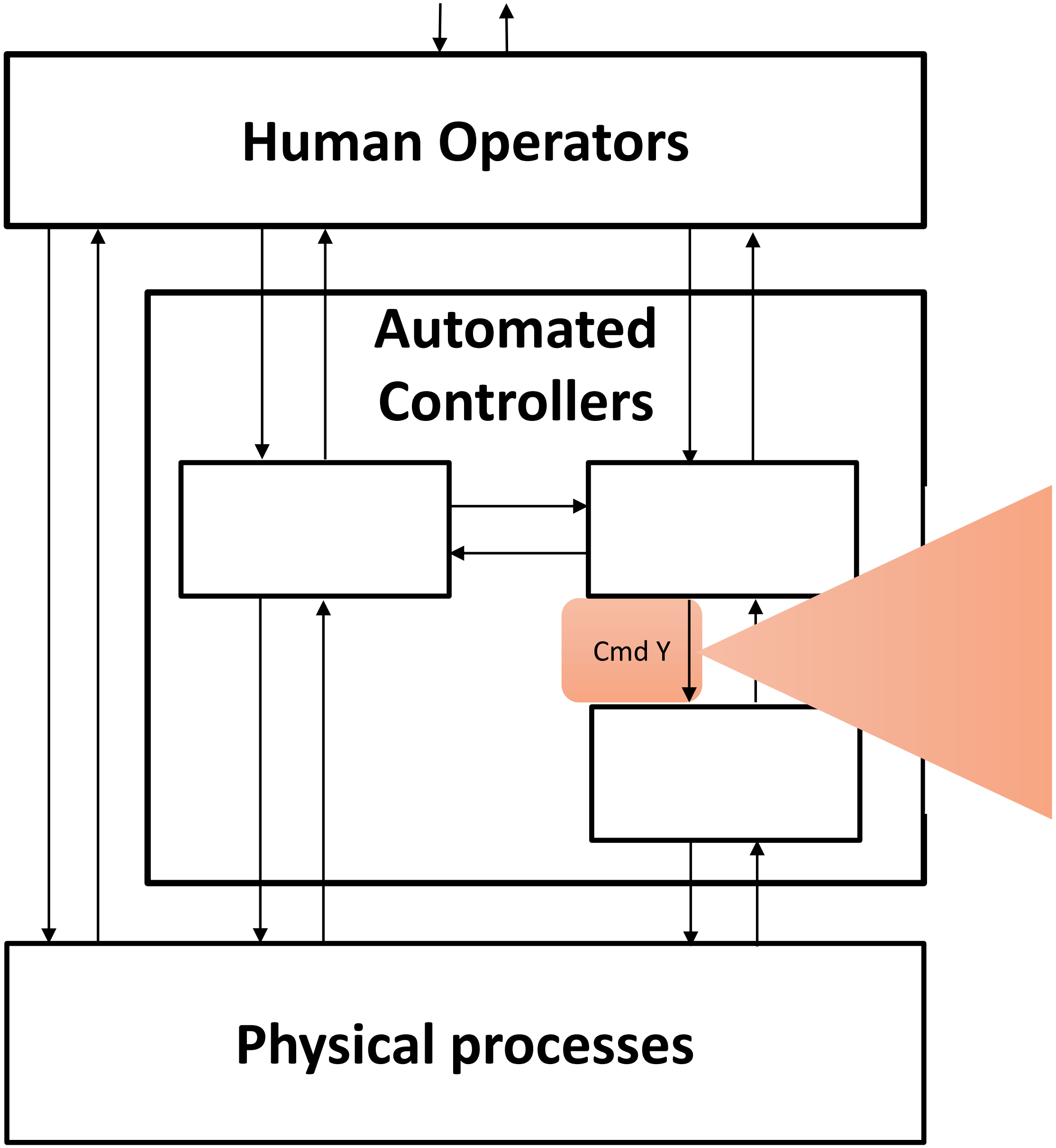
Define System boundary

Environment

System

(Leveson and Thomas, 2018)

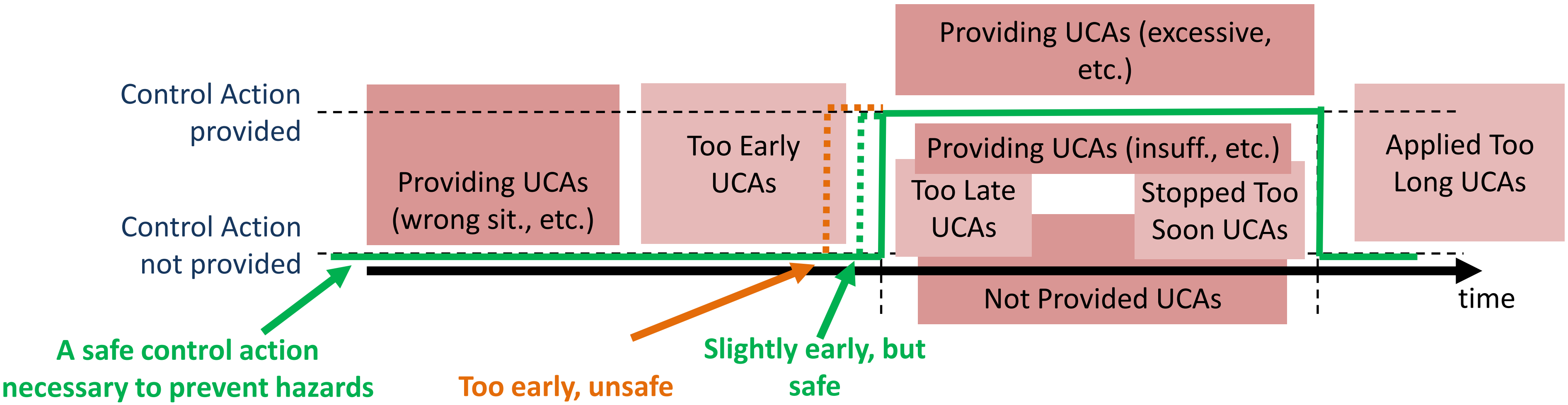
STPA: Identify Unsafe Control Actions (UCA)



Not provided causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long

(Thomas, 2017)

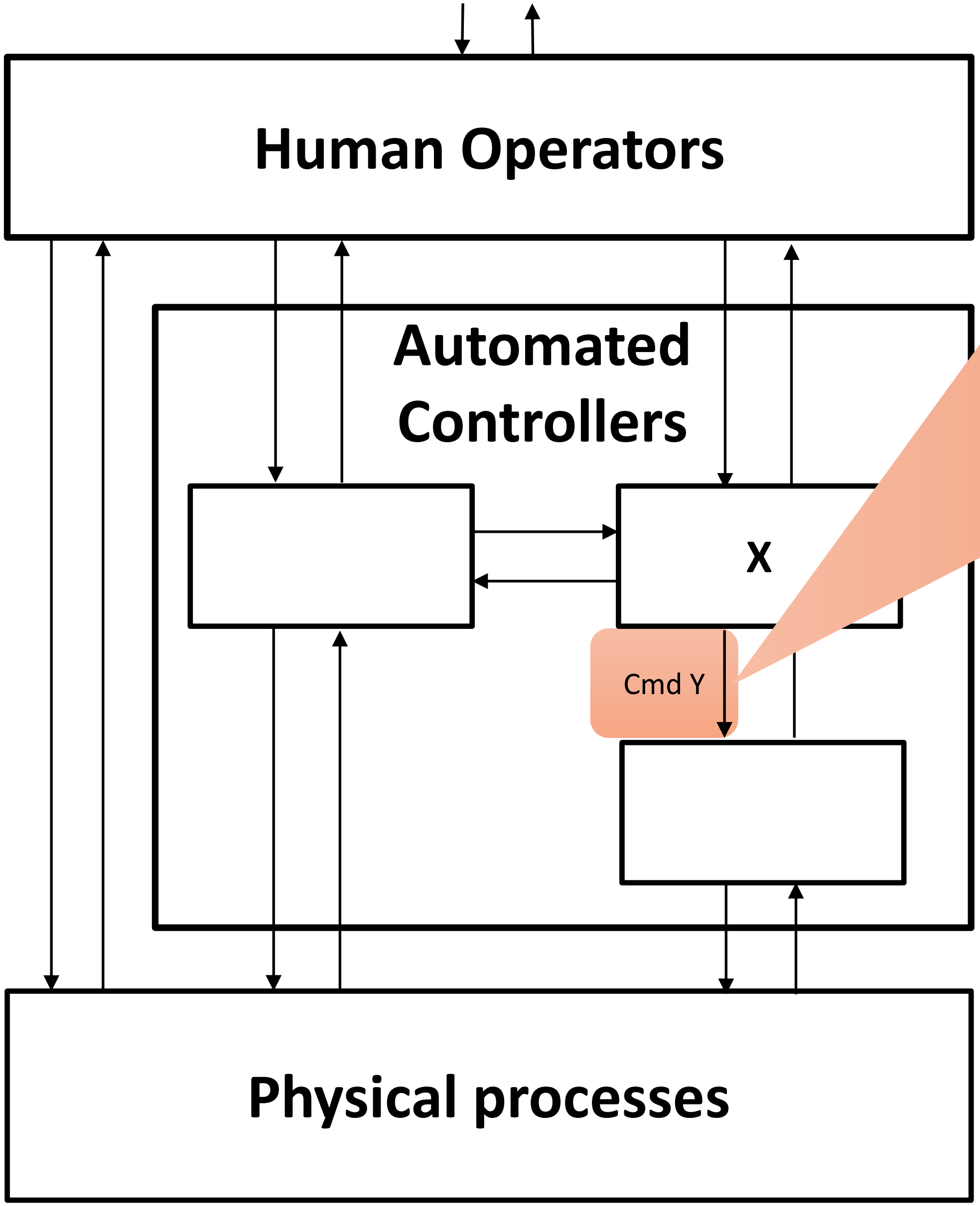
STPA UCA Bounding


Control Action provided
Control Action not provided
Providing UCAs (wrong sit., etc.)
Too Early UCAs
Providing UCAs (excessive, etc.)
Providing UCAs (insuff., etc.)
Too Late UCAs
Stopped Too Soon UCAs
Not Provided UCAs
Applied Too Long UCAs
time
A safe control action necessary to prevent hazards
Too early, unsafe
Slightly early, but safe

	Not providing causes hazard	Providing causes hazard *[in wrong situation, excessive, insufficient, repetitive, wrong direction, etc.]*	Too early, too late, Order	Stopped Too Soon / Applied too long
Control Actions	?	?	?	?

The complete set of UCAs will fully bound the necessary safe behavior

STPA: Identify Unsafe Control Actions (UCA)



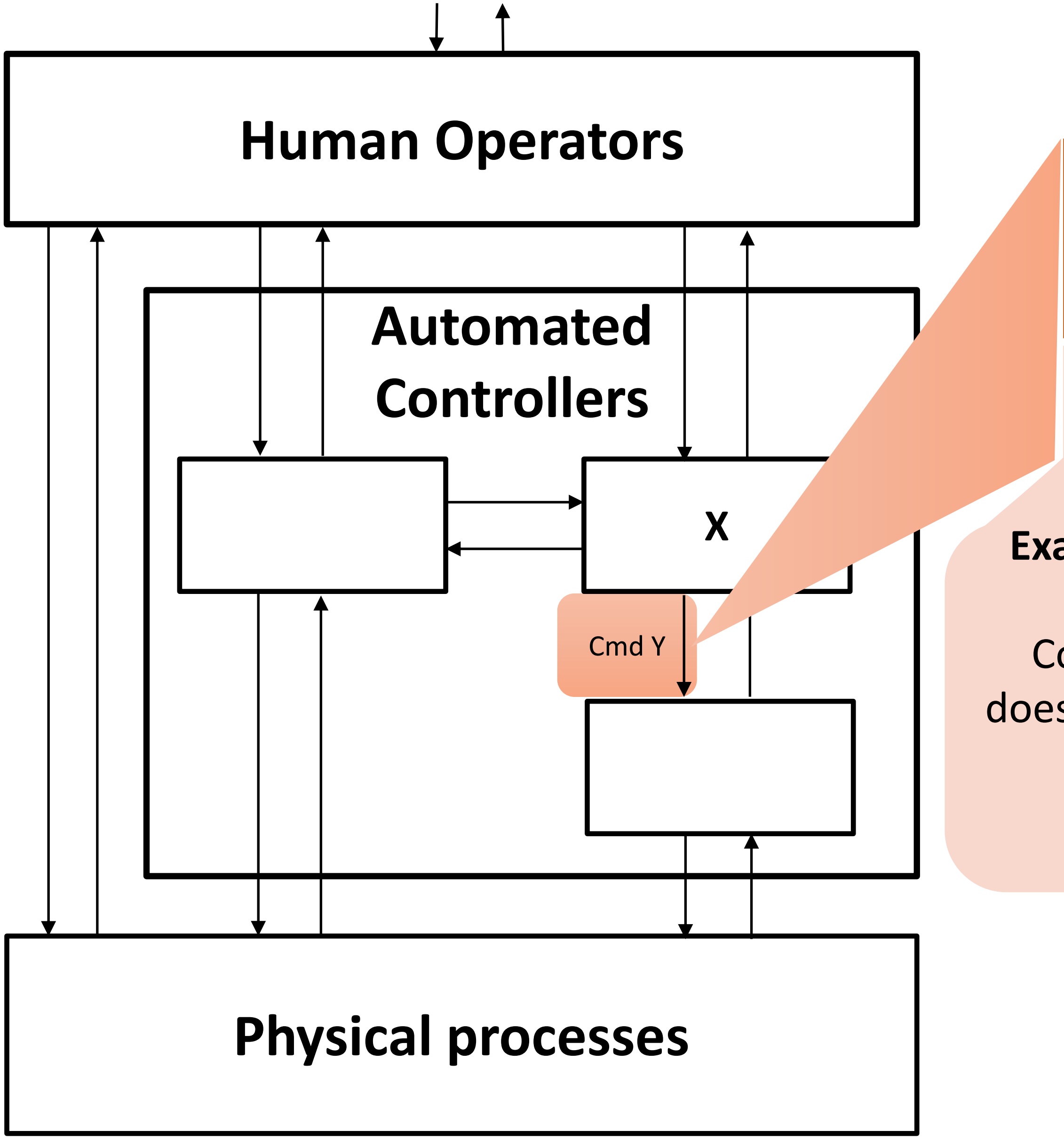
Not provided causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long

Example UCA:

Controller X
does not provide
Cmd Y
when Z
[H-1]

(Thomas, 2017)

STPA: Identify Unsafe Control Actions (UCA)

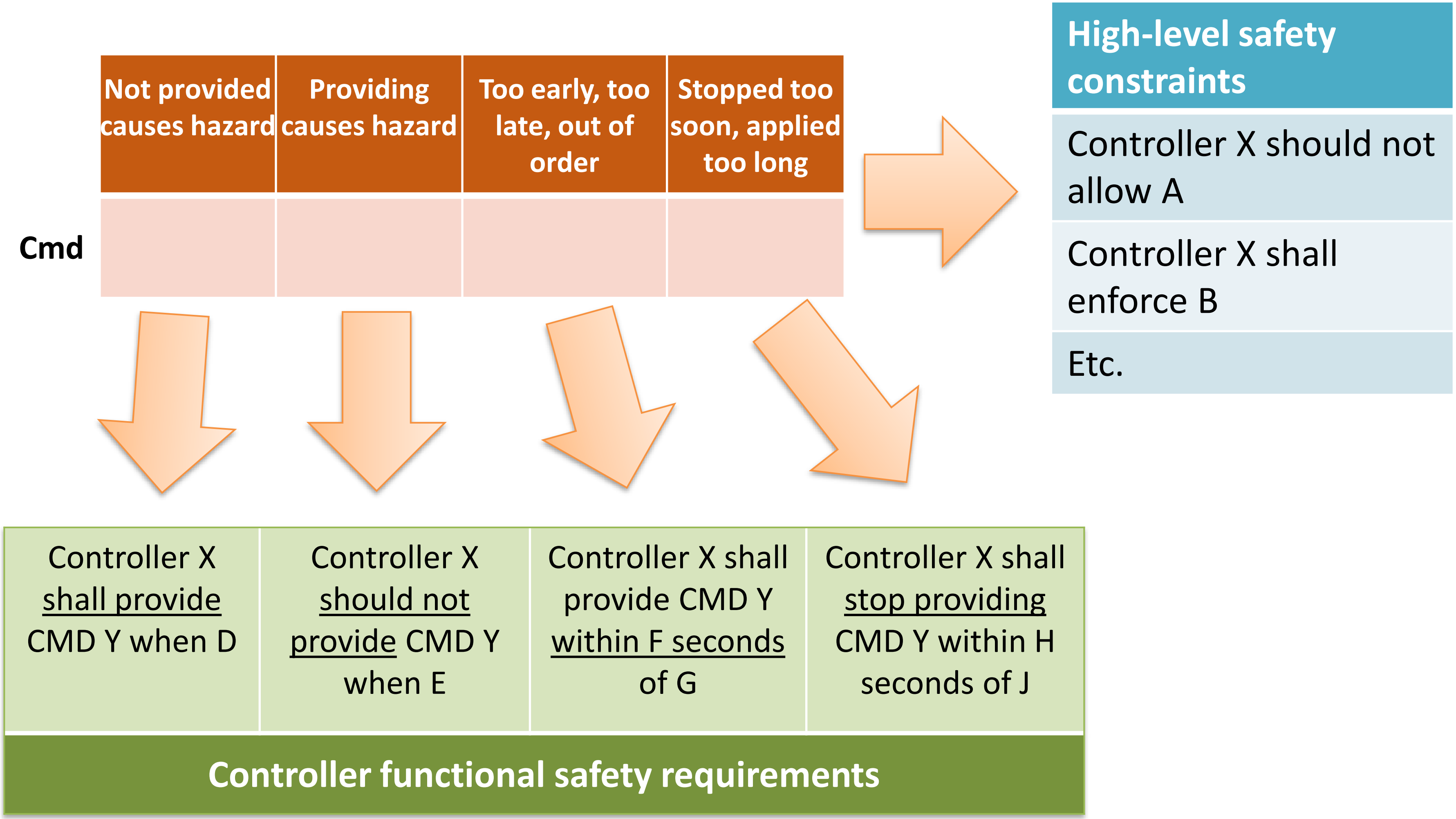


Not provided causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long

Example UCA:	UCA Syntax:
Controller X	Source Cont.
does not provide	UCA Type
Cmd Y	Control Action
when Z	Context
[H-1]	Traceability

(Thomas, 2017)

Generating constraints and requirements



	Not provided causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Cmd				

High-level safety constraints
Controller X should not allow A
Controller X shall enforce B
Etc.

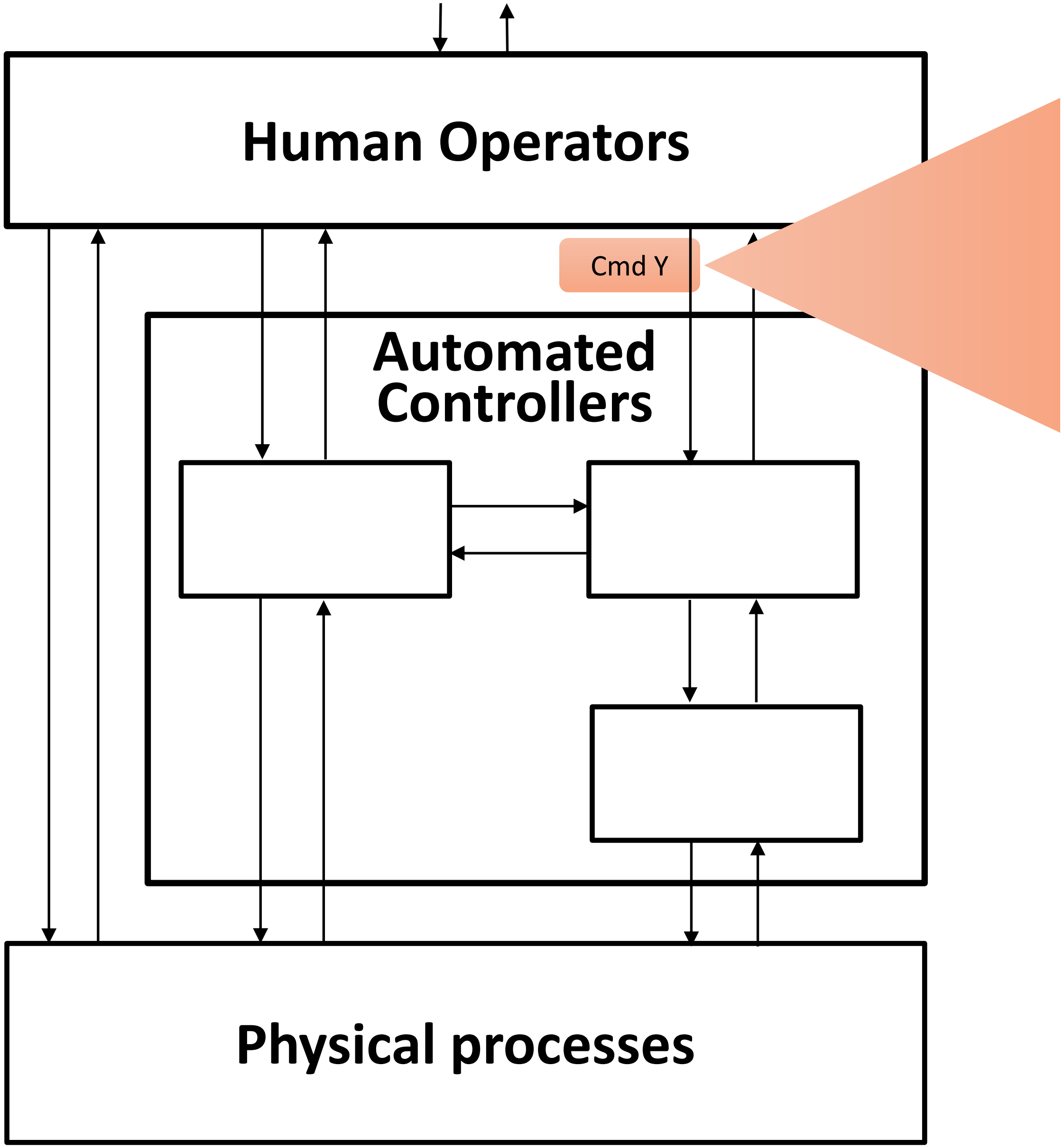
Controller X shall provide CMD Y when D	Controller X should not provide CMD Y when E	Controller X shall provide CMD Y within F seconds of G	Controller X shall stop providing CMD Y within H seconds of J

Controller functional safety requirements

(Thomas, 2017)

What about human interactions?

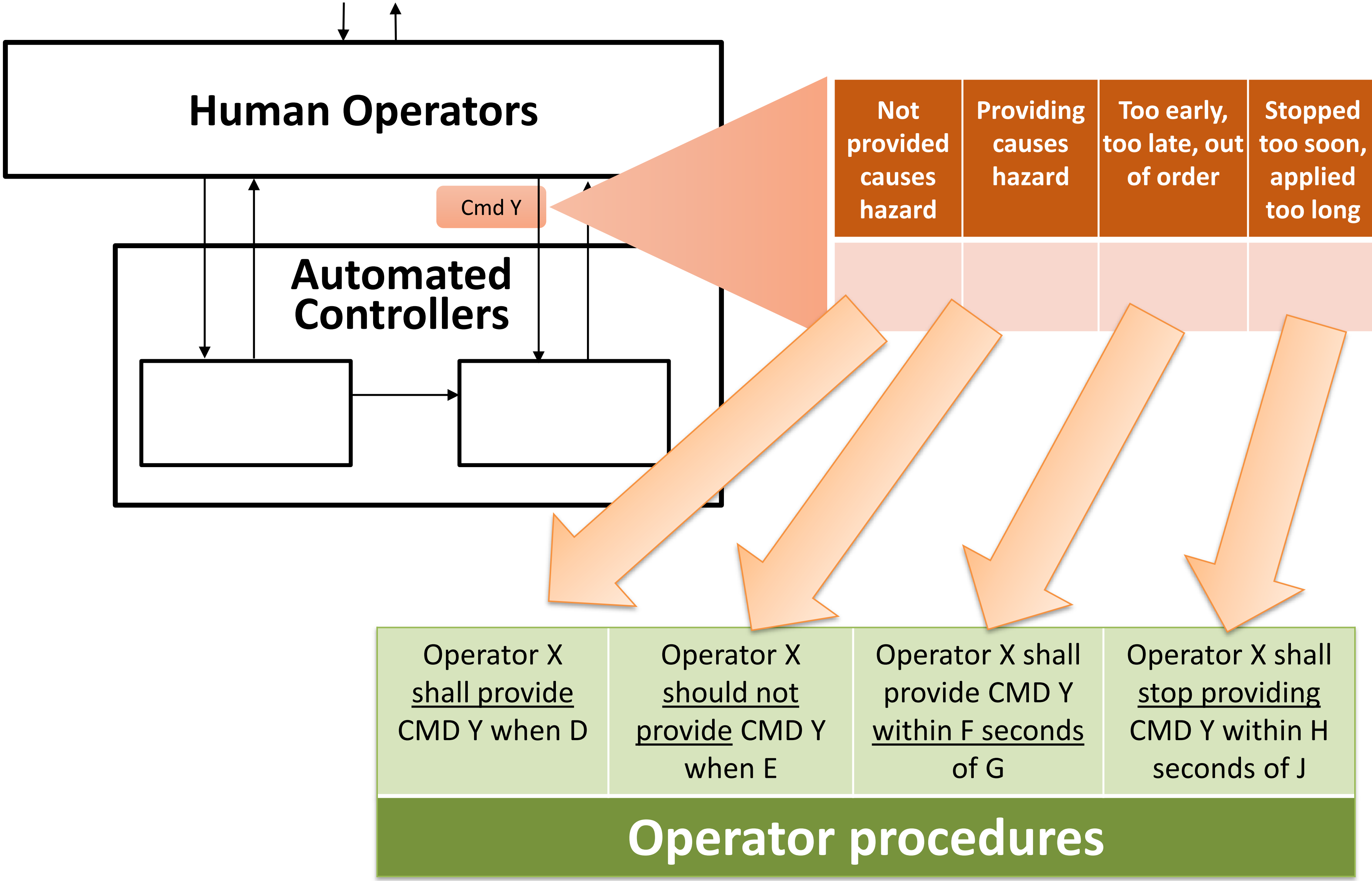
Unsafe Control Actions (UCA)



Not provided causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long

(Thomas, 2017)

Generating & validating operator procedures

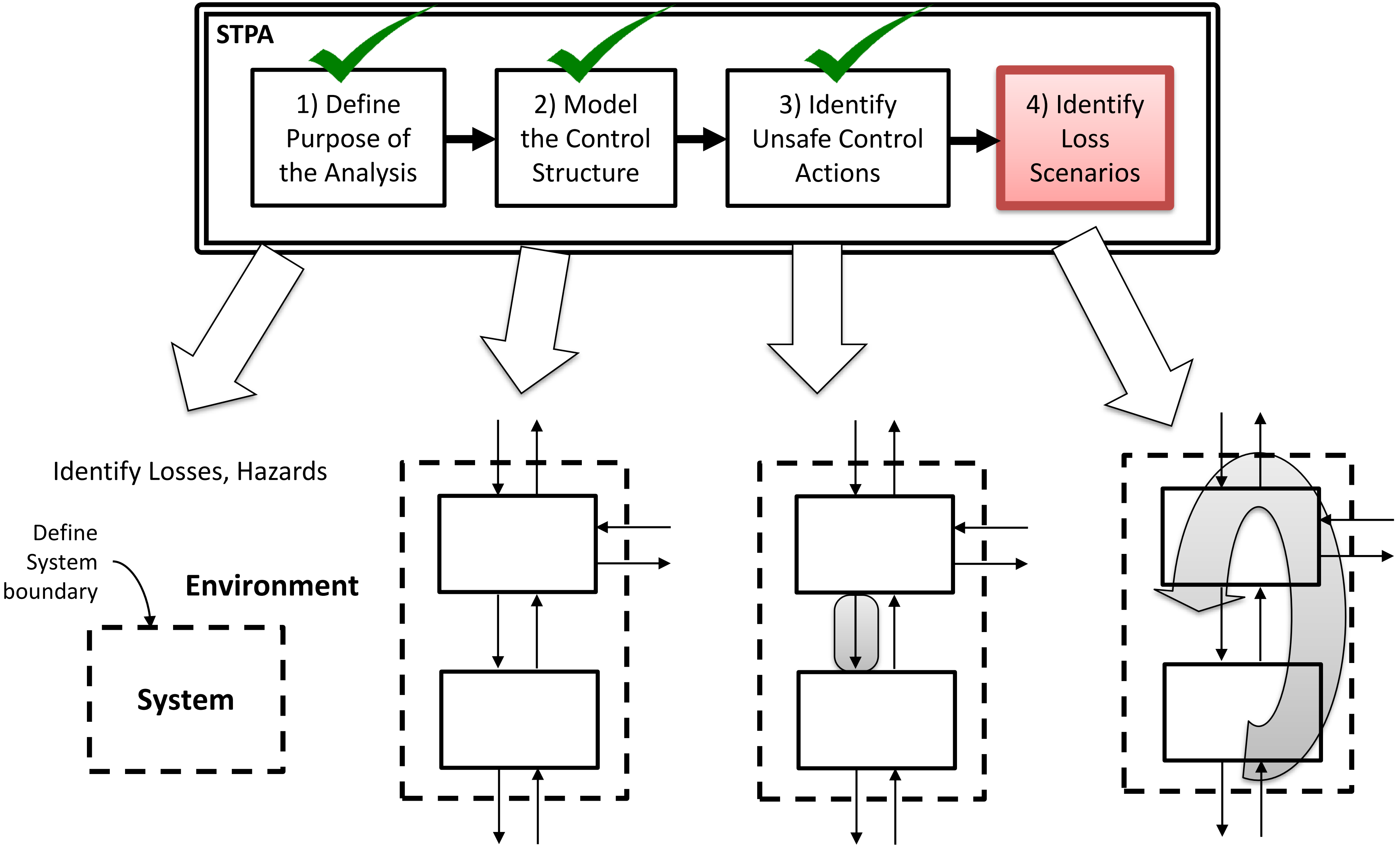


Not provided causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long

Operator X shall provide CMD Y when D	Operator X should not provide CMD Y when E	Operator X shall provide CMD Y within F seconds of G	Operator X shall stop providing CMD Y within H seconds of J

Operator procedures

(John Thomas, 2017)



STPA

1) Define Purpose of the Analysis

2) Model the Control Structure

3) Identify Unsafe Control Actions

4) Identify Loss Scenarios

Identify Losses, Hazards

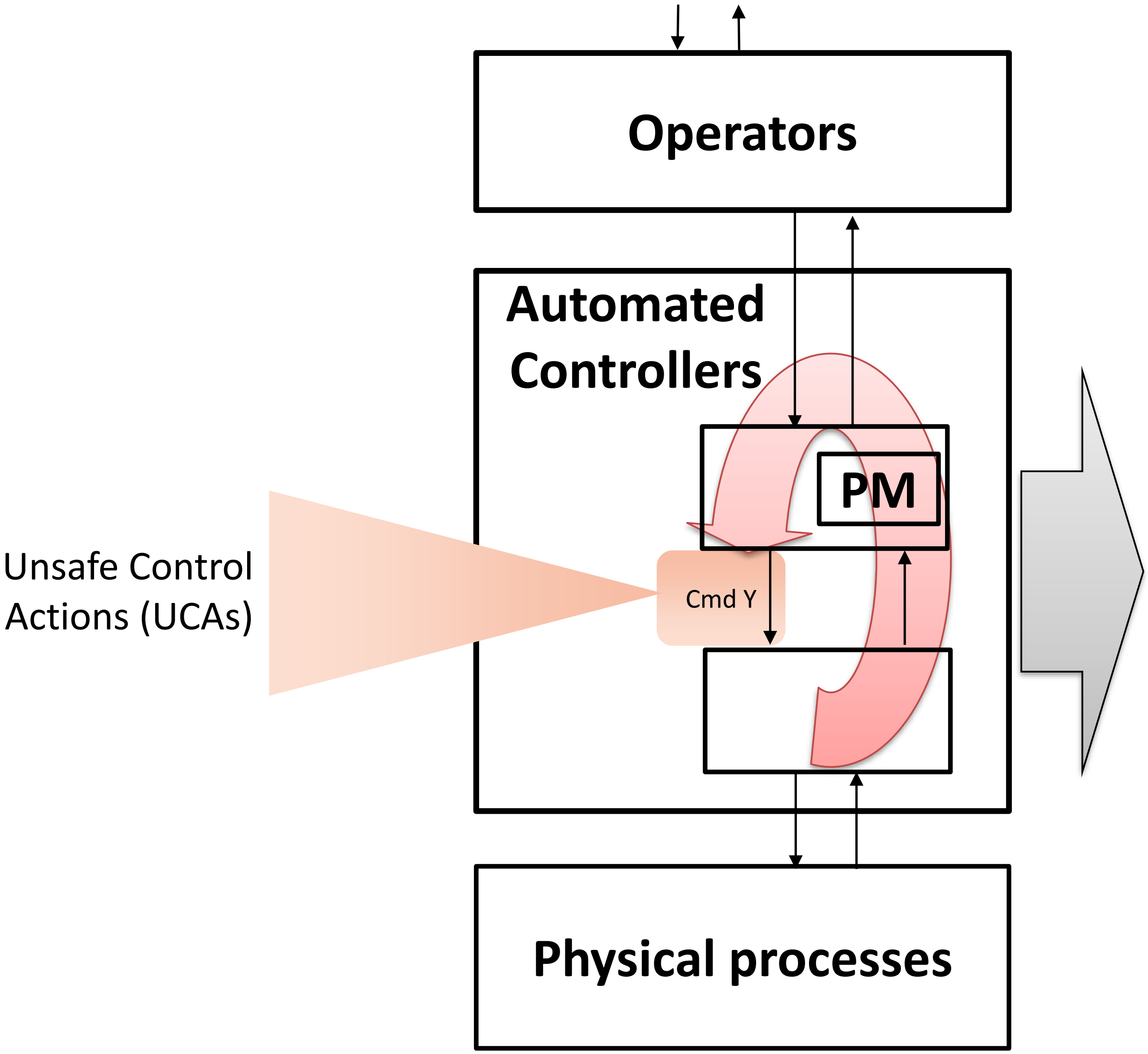
Define System boundary

Environment

System

(Leveson and Thomas, 2018)

Identify loss scenarios

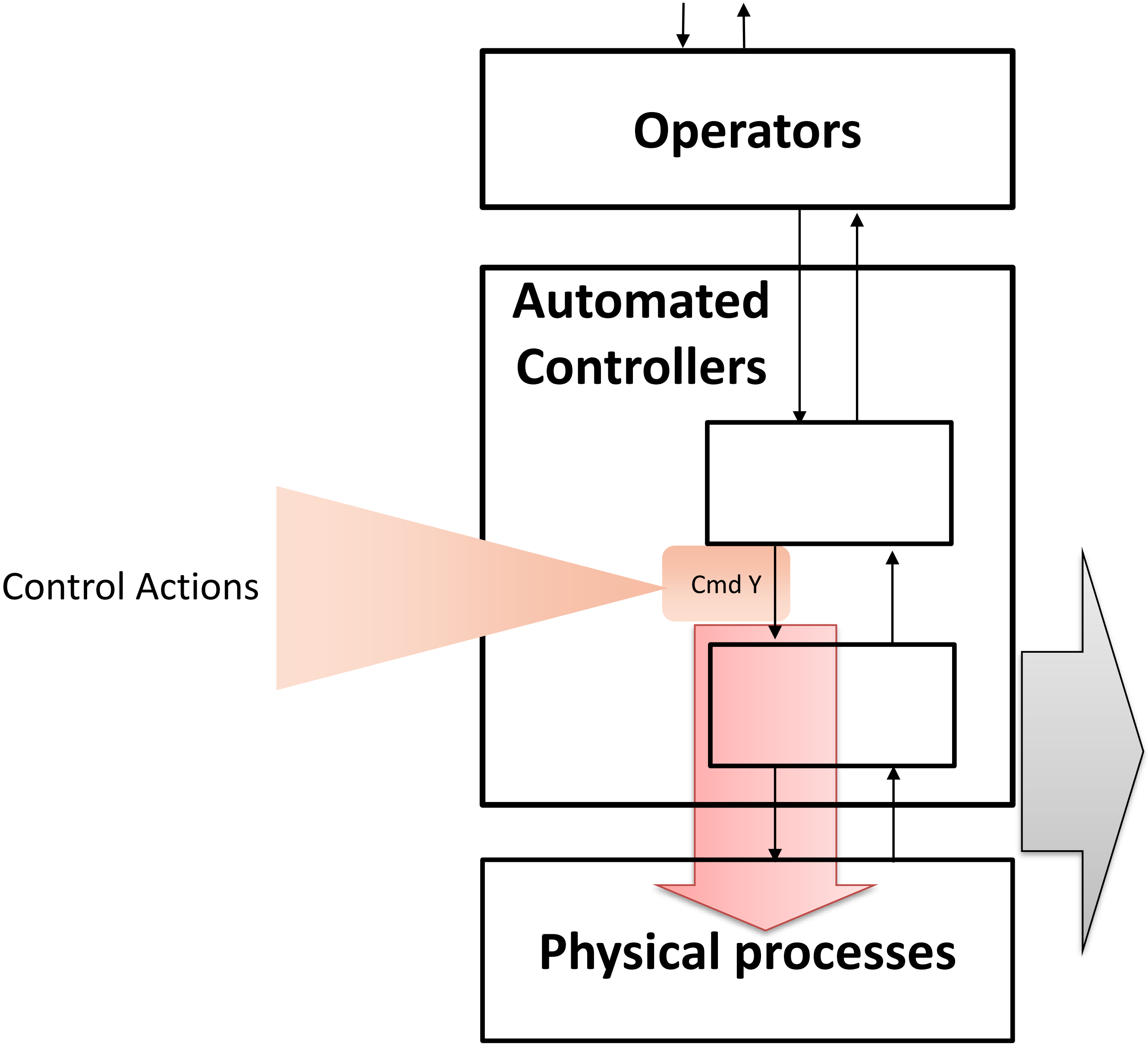


What could cause Unsafe Control Actions?

Scenarios
Controller incorrectly believes A because …
Controller control algorithm does not enforce B because …
Incorrect feedback C received because …
Sensor failure causes…
Etc.

(John Thomas, 2017)

Identify loss scenarios



How could control actions not be executed or followed properly?

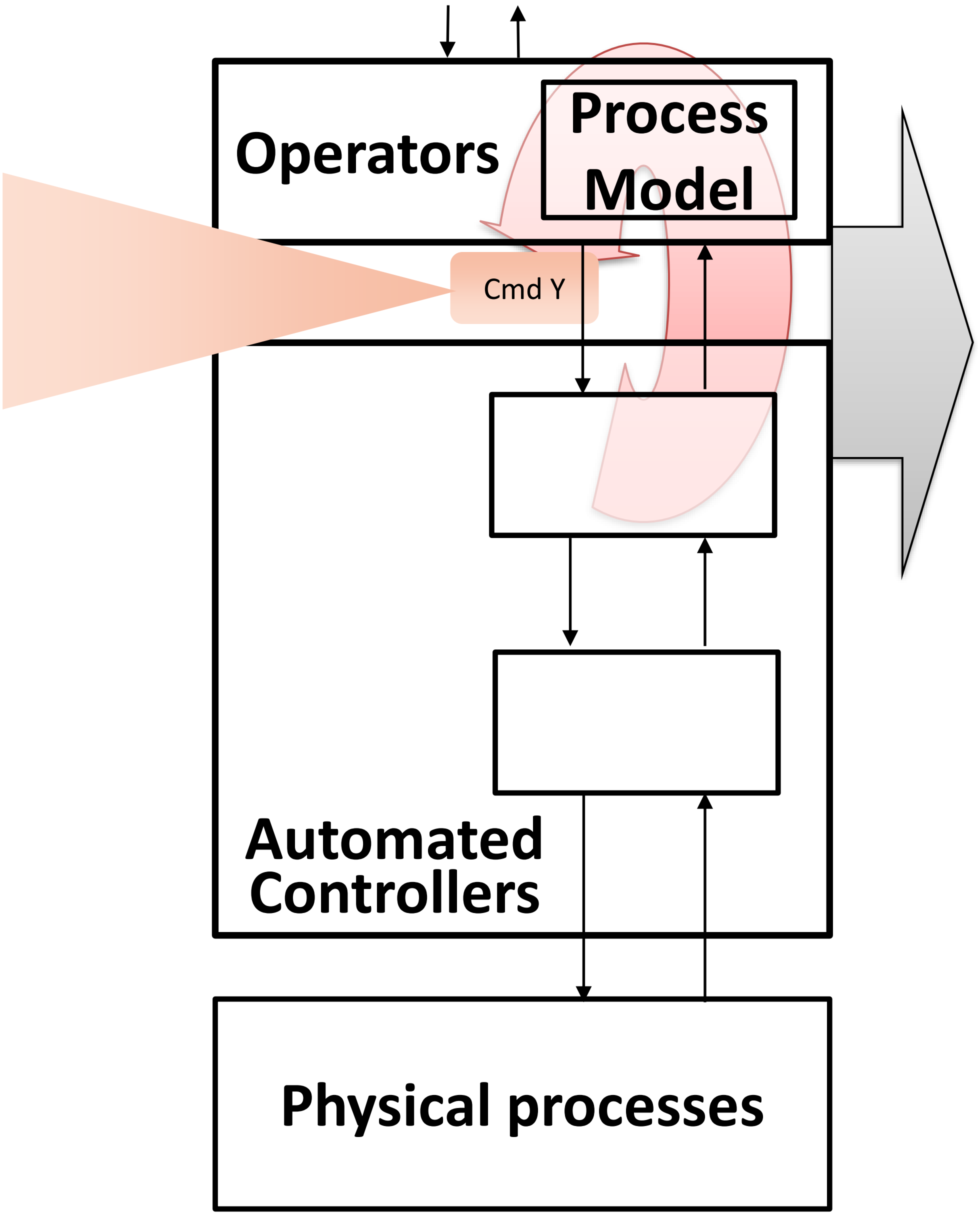
Scenarios
Cmd sent but not received because…
Cmd received but ignored because…
Actuator failure causes…

(Thomas, 2017)

What about human interactions?

Identify loss scenarios

Unsafe Control Actions (UCAs)

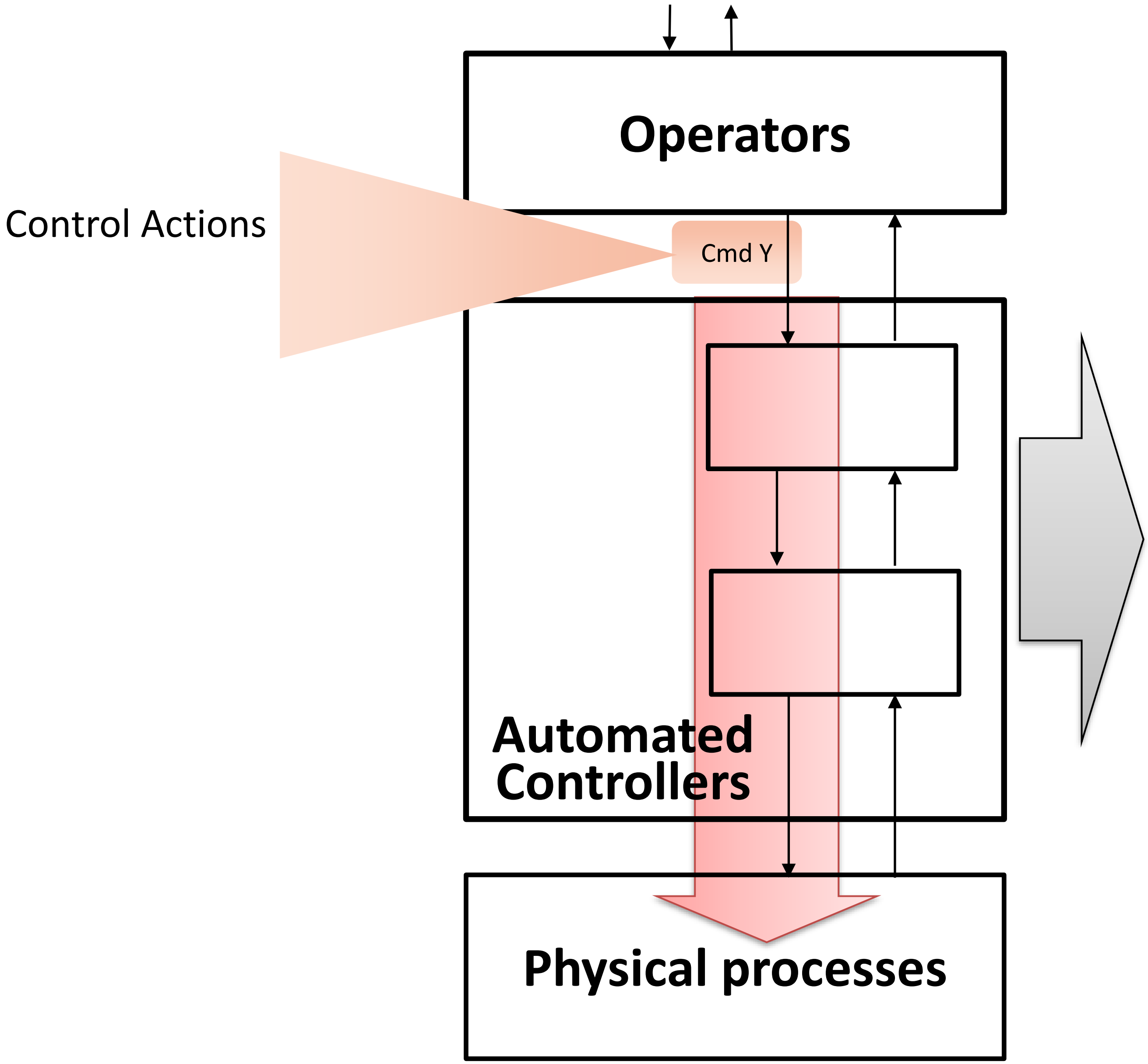


What could cause Unsafe Control Actions?

Scenarios
Op responded to failure in A by …
Op incorrectly believes B because …
Op does not perform C because …
Op received incorrect feedback D because …
Etc.

(John Thomas, 2017)

Identify loss scenarios

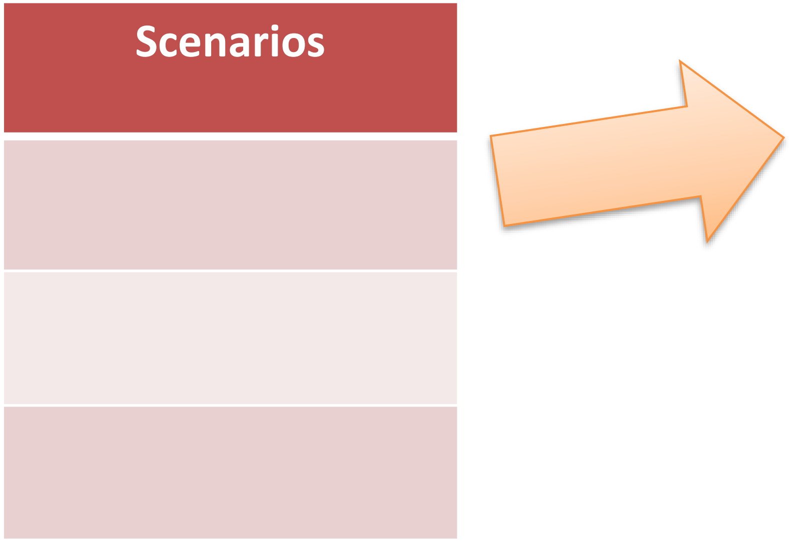


How could control actions not be executed or followed properly?

Scenarios
Op cmd sent but not received because…
Op cmd received but ignored because…
Actuator failure causes…

(Thomas, 2017)

Provide Solutions



Scenarios

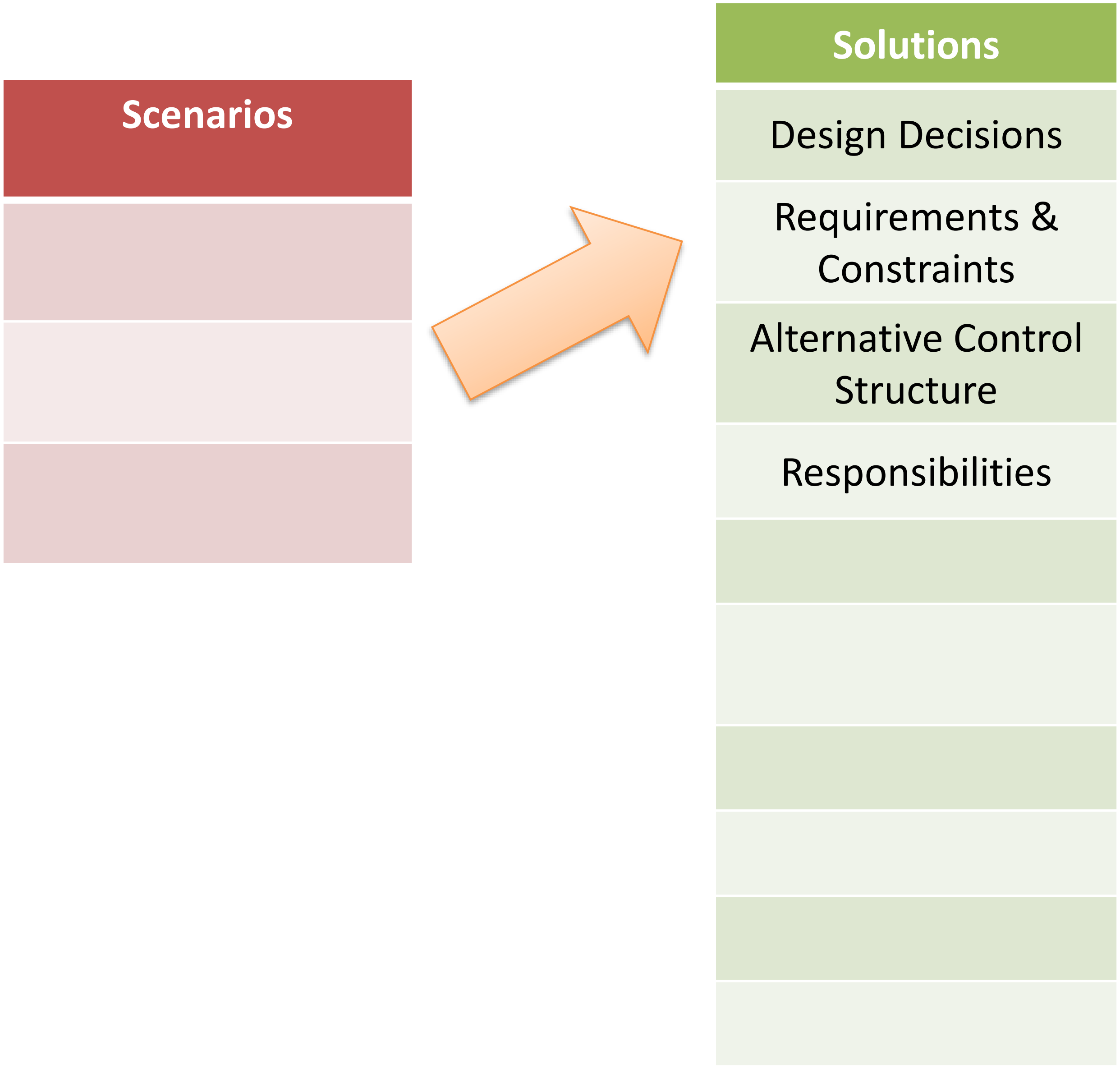
Solutions
Component A must be able to respond within B seconds to avoid C
Controller X must provide D when E to prevent F
Component G shall automatically operate within H seconds when J
Operator must provide K and L together when M to prevent N (assumption)
Etc.

Rationale and assumptions identified

Every recommendation and requirement is traceable

(Thomas, 2017)

Provide Solutions



Scenarios

Solutions

Design Decisions

Requirements & Constraints

Alternative Control Structure

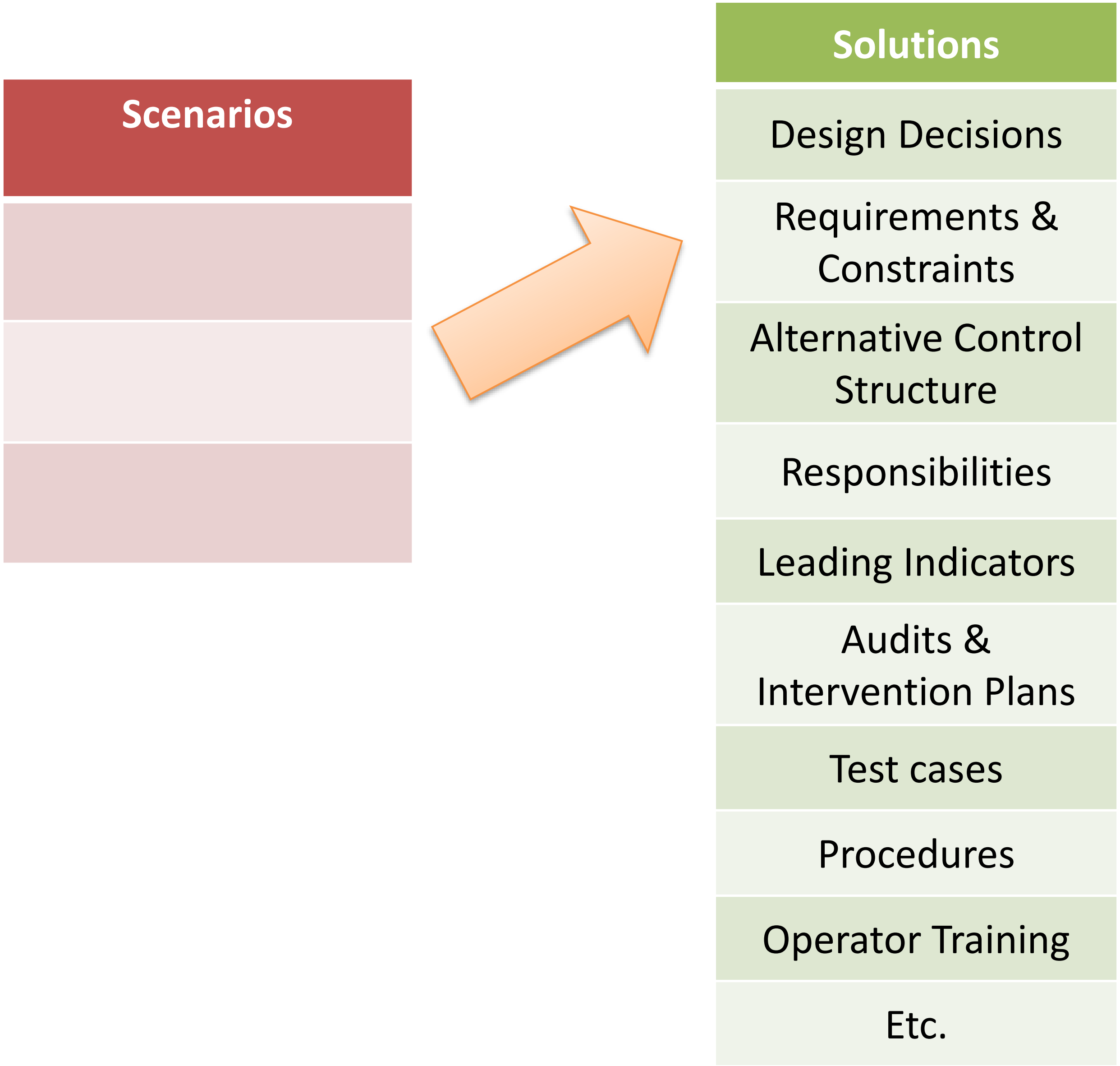
Responsibilities

Rationale and assumptions identified

Every recommendation and decision is traceable

(Thomas, 2017)

Provide Solutions



Scenarios

Solutions
Design Decisions
Requirements & Constraints
Alternative Control Structure
Responsibilities
Leading Indicators
Audits & Intervention Plans
Test cases
Procedures
Operator Training
Etc.

Rationale and assumptions identified

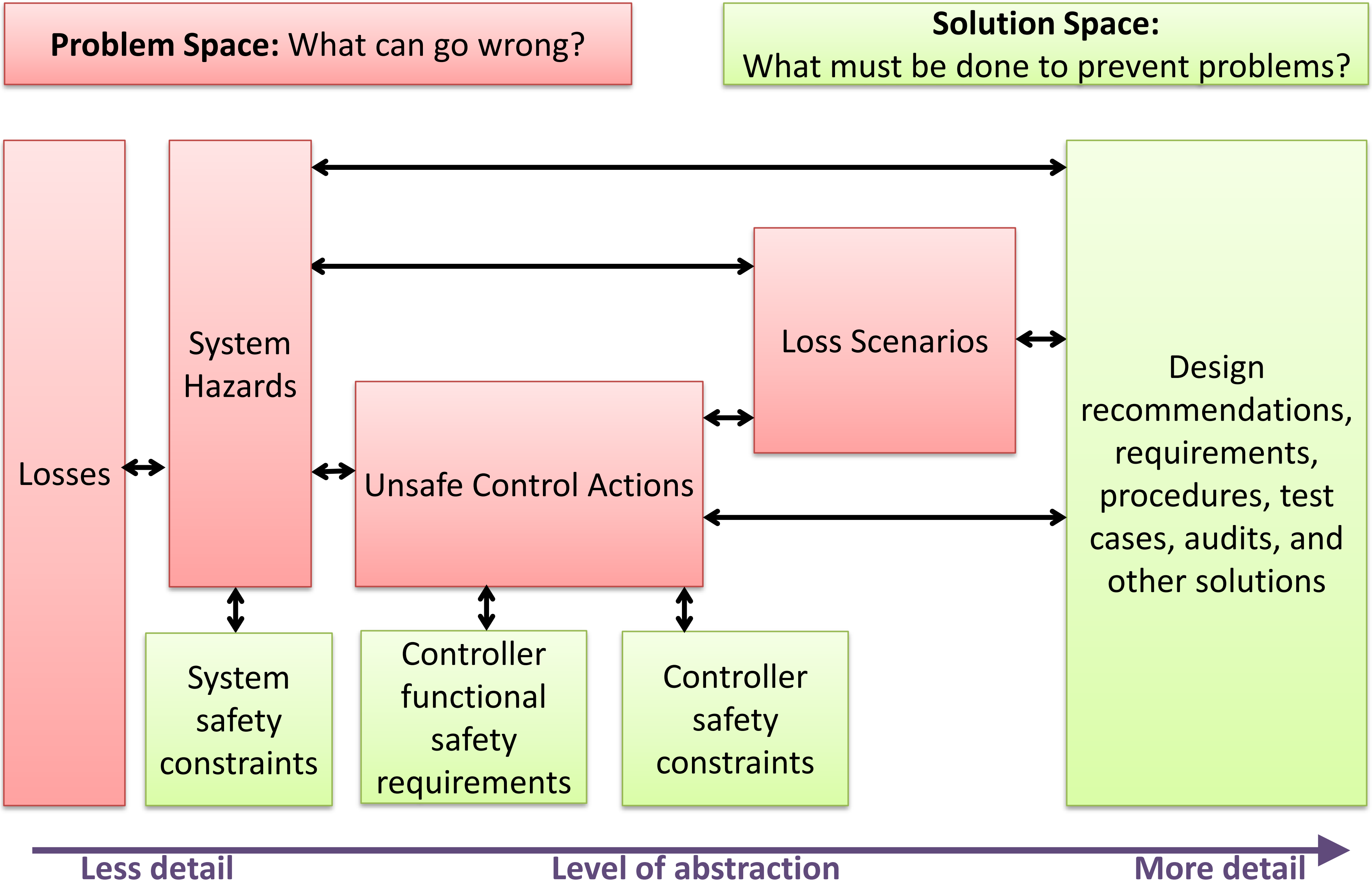
Every recommendation and decision is traceable

(Thomas, 2017)

STPA Outputs

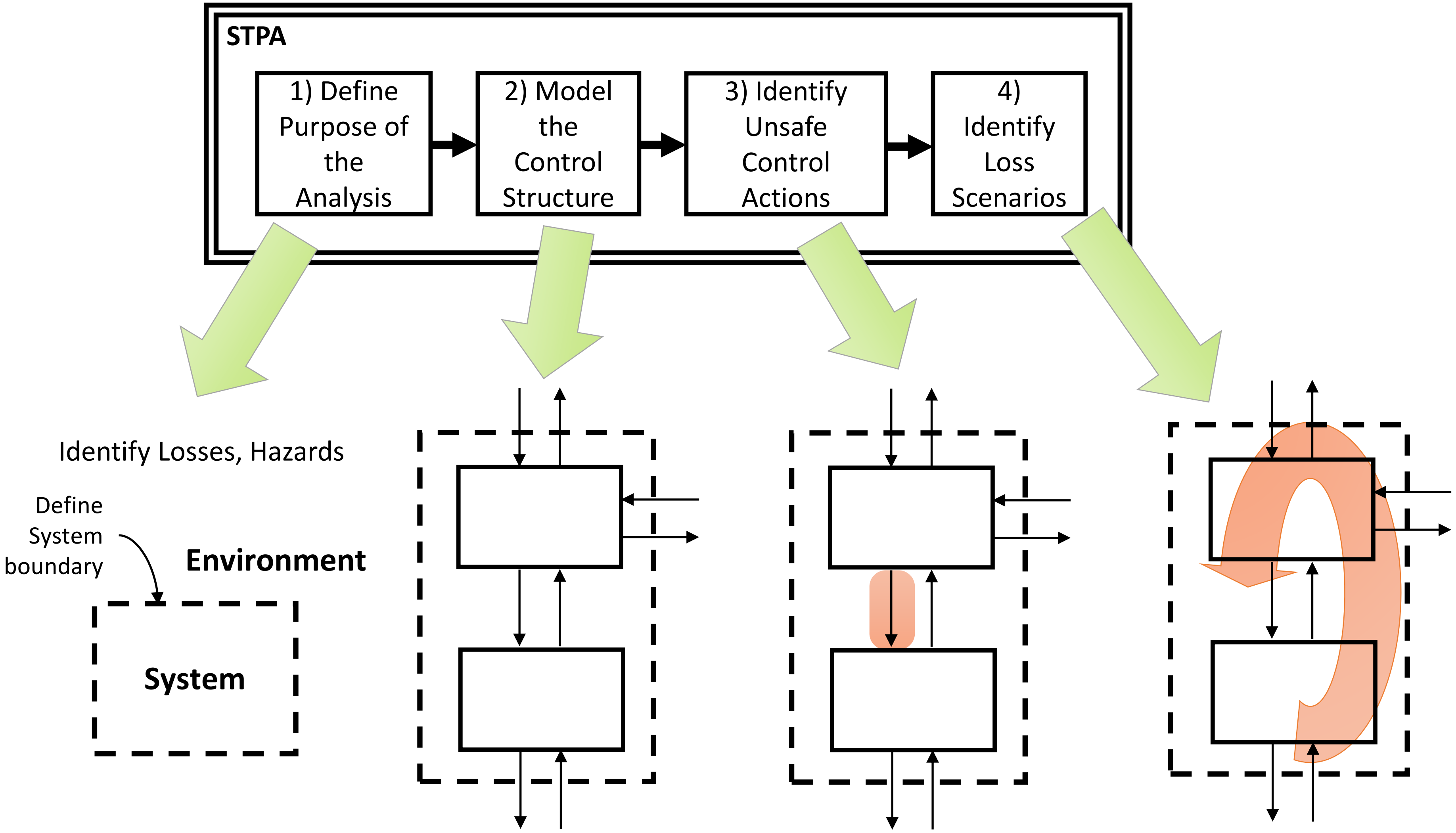
- Loss **scenarios**
- **Constraints** that need to be enforced
- A **conceptual architecture** that enforces the constraints
- The **responsibilities** that need to be allocated
- **Assumptions** that need to be validated
- Behavioral **requirements** that need to be enforced
- **Procedures**
- Critical **test scenarios** / test cases
- Operational **leading indicators** of risk
- Audit plan
- Etc.

STPA: Traceability is maintained throughout


Problem Space: What can go wrong?
Solution Space:
What must be done to prevent problems?
Losses
System Hazards
Unsafe Control Actions
Loss Scenarios
Design recommendations, requirements, procedures, test cases, audits, and other solutions
System safety constraints
Controller functional safety requirements
Controller safety constraints
Less detail
Level of abstraction
More detail

(Thomas, 2017)

STPA Overview



STPA

1) Define Purpose of the Analysis

2) Model the Control Structure

3) Identify Unsafe Control Actions

4) Identify Loss Scenarios

Identify Losses, Hazards

Define System boundary

Environment

System

(Leveson and Thomas, 2018)