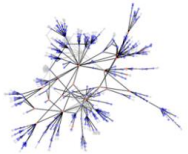


STPA Advanced Tutorial

STPA Workshop
Cambridge, MA
26 March 2013

Cody Fleming
Massachusetts Institute of Technology

Agenda

- STAMP/STPA Background
- Exercises
- Advanced Topics
- Discussion
  - Participant questions
  - Current Research Trends

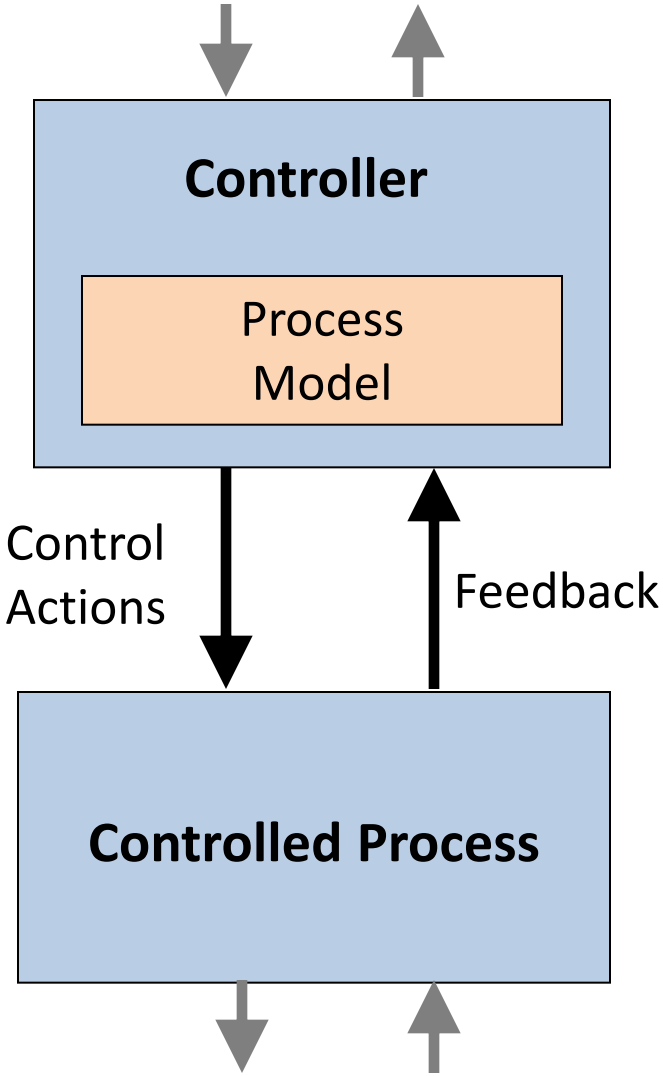
Systems approach to safety eng

STAMP Model

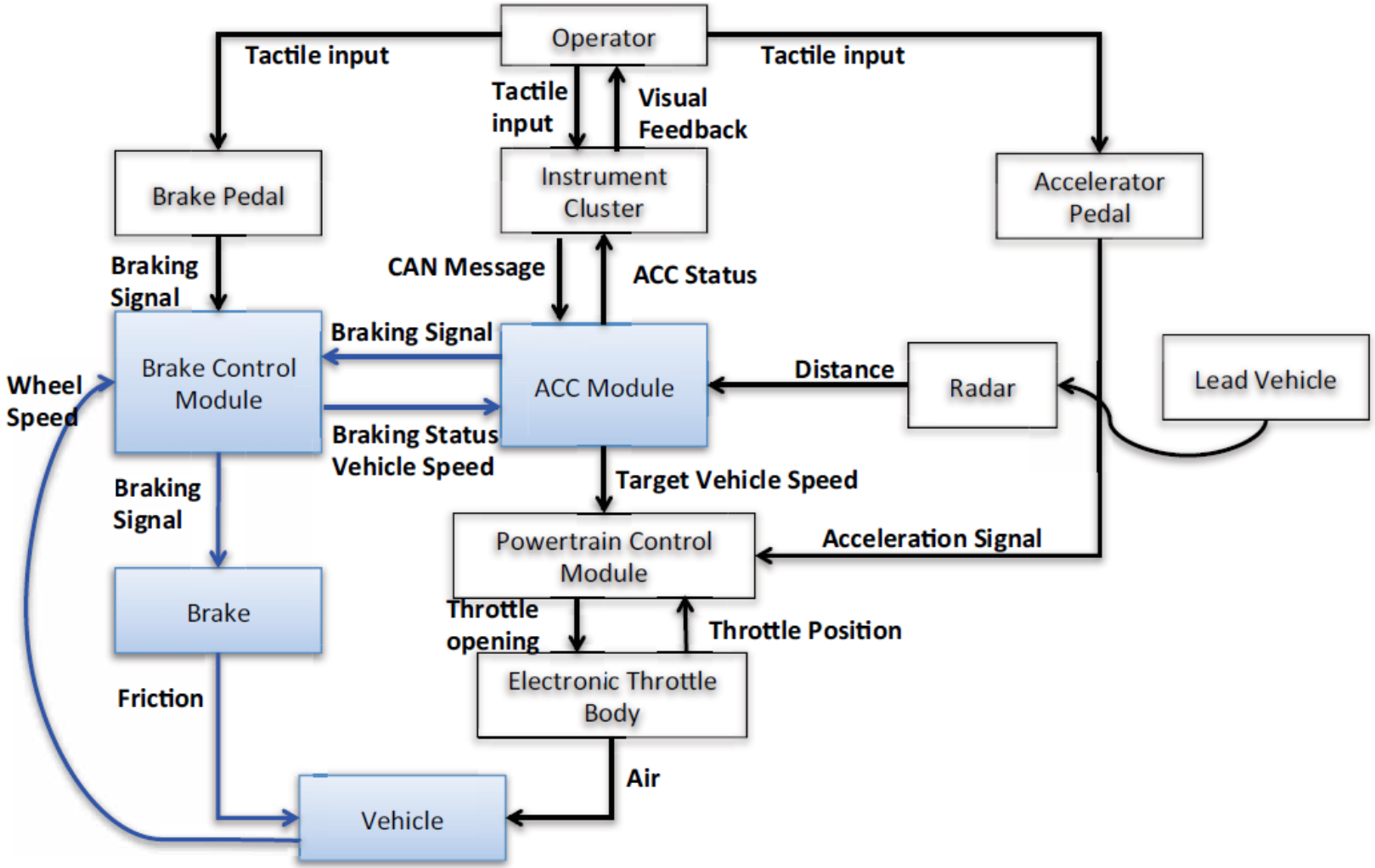
- Accidents are more than a chain of events, they involve complex dynamic **processes**.
- Treat accidents as a **control problem**, not a failure problem
- Prevent accidents by enforcing constraints on component behavior and **interactions**
- Captures more causes of accidents:
  - Component failure accidents
  - Unsafe interactions among components
  - Complex human, software behavior
  - Design errors
  - Flawed requirements
    - esp. software-related accidents

(Leveson, 2003); (Leveson, 2011)

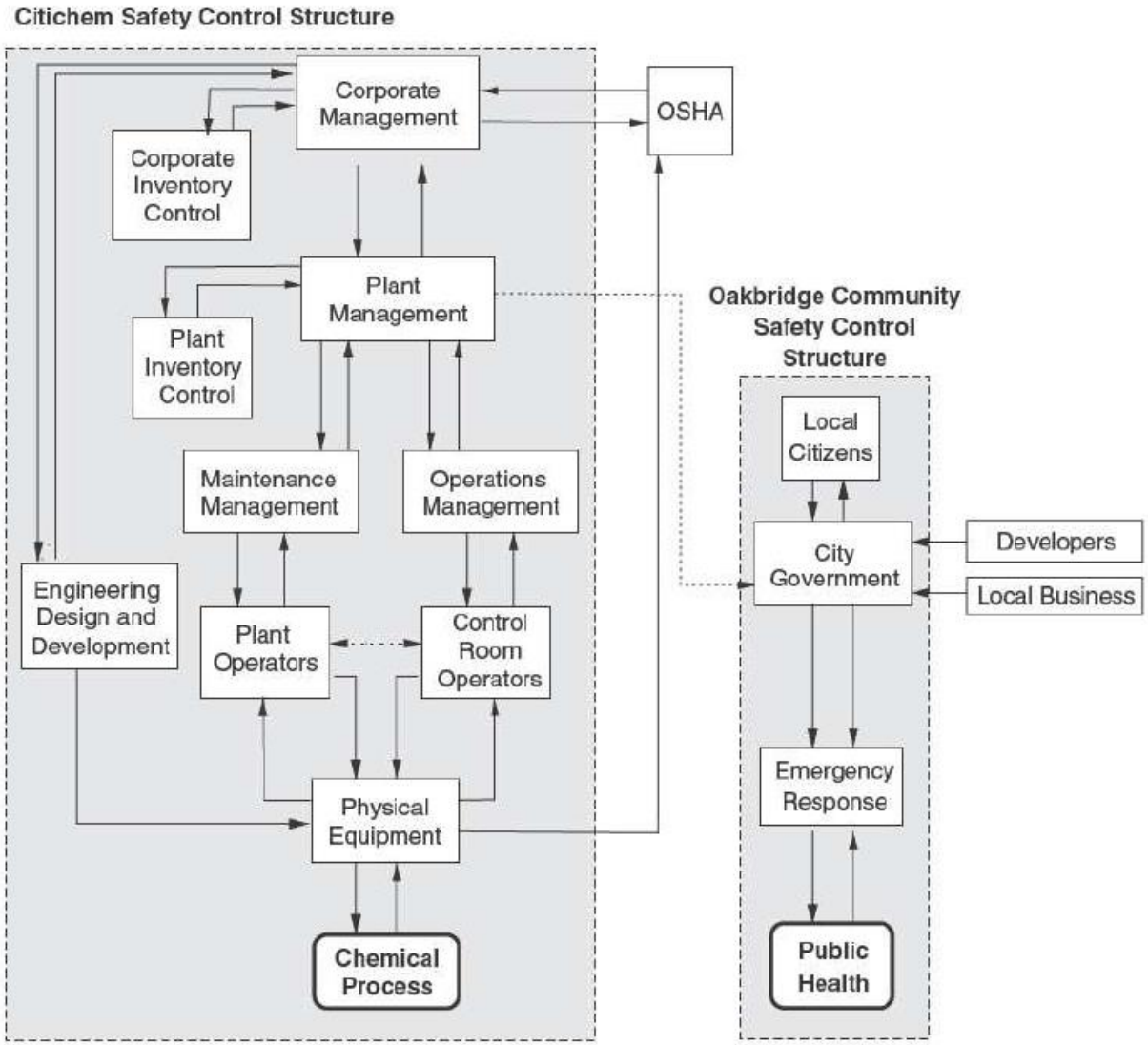
Basic Control Loop



Example: ACC – BCM Control Loop

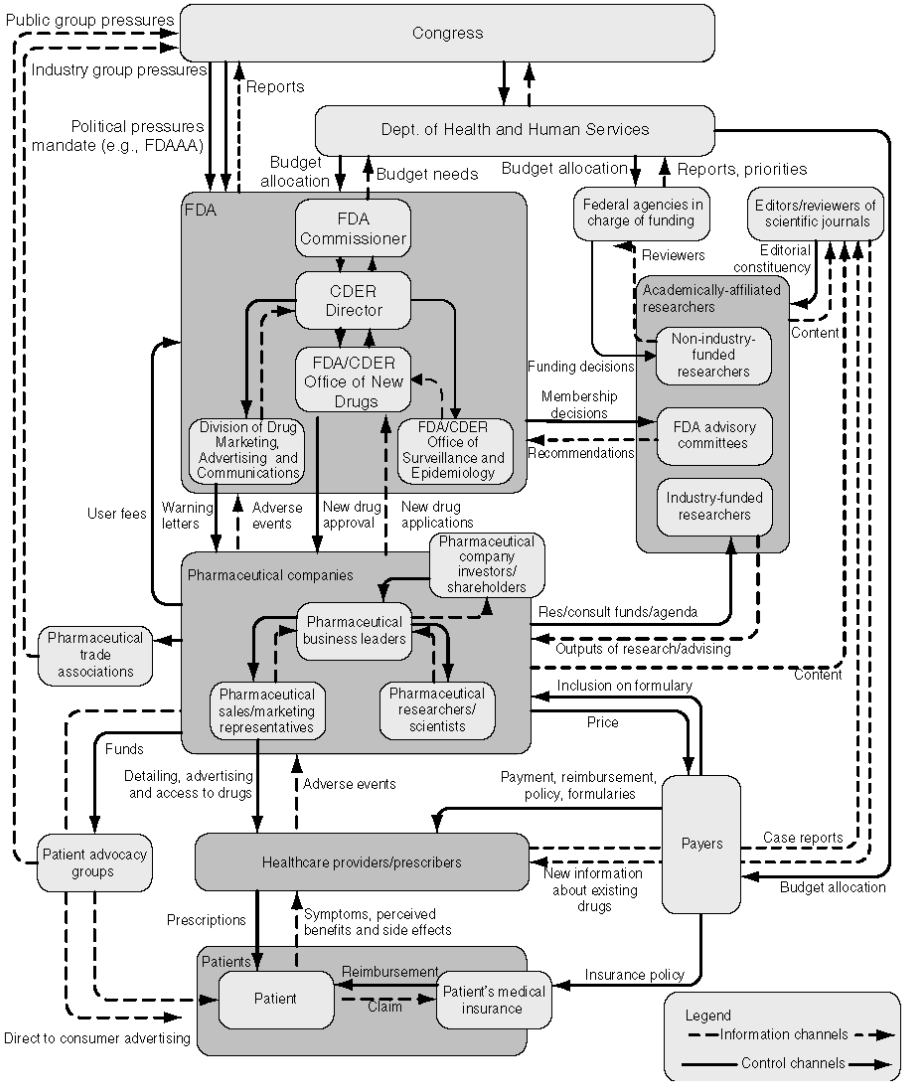


Safety Control Structure



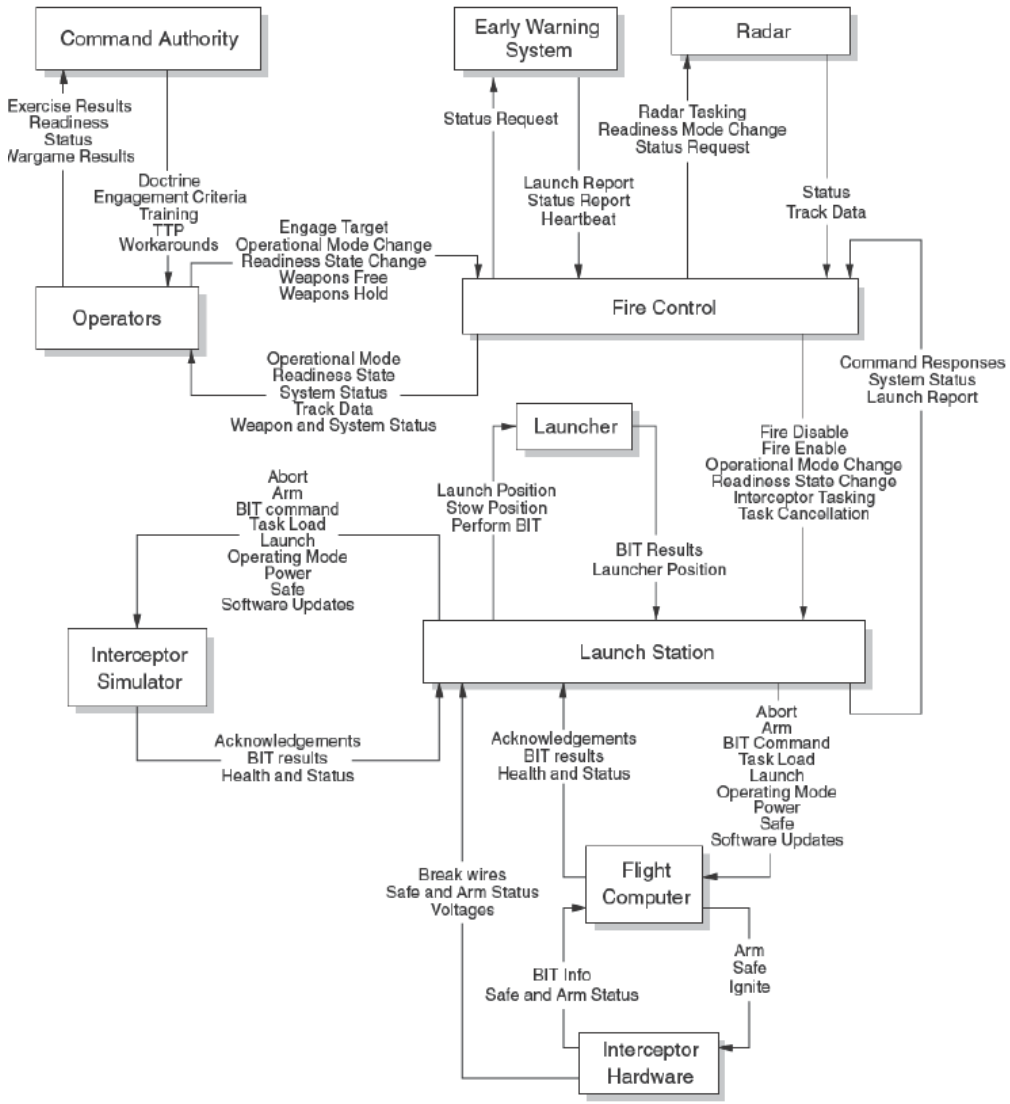
ESW p354

Safety Control Structure



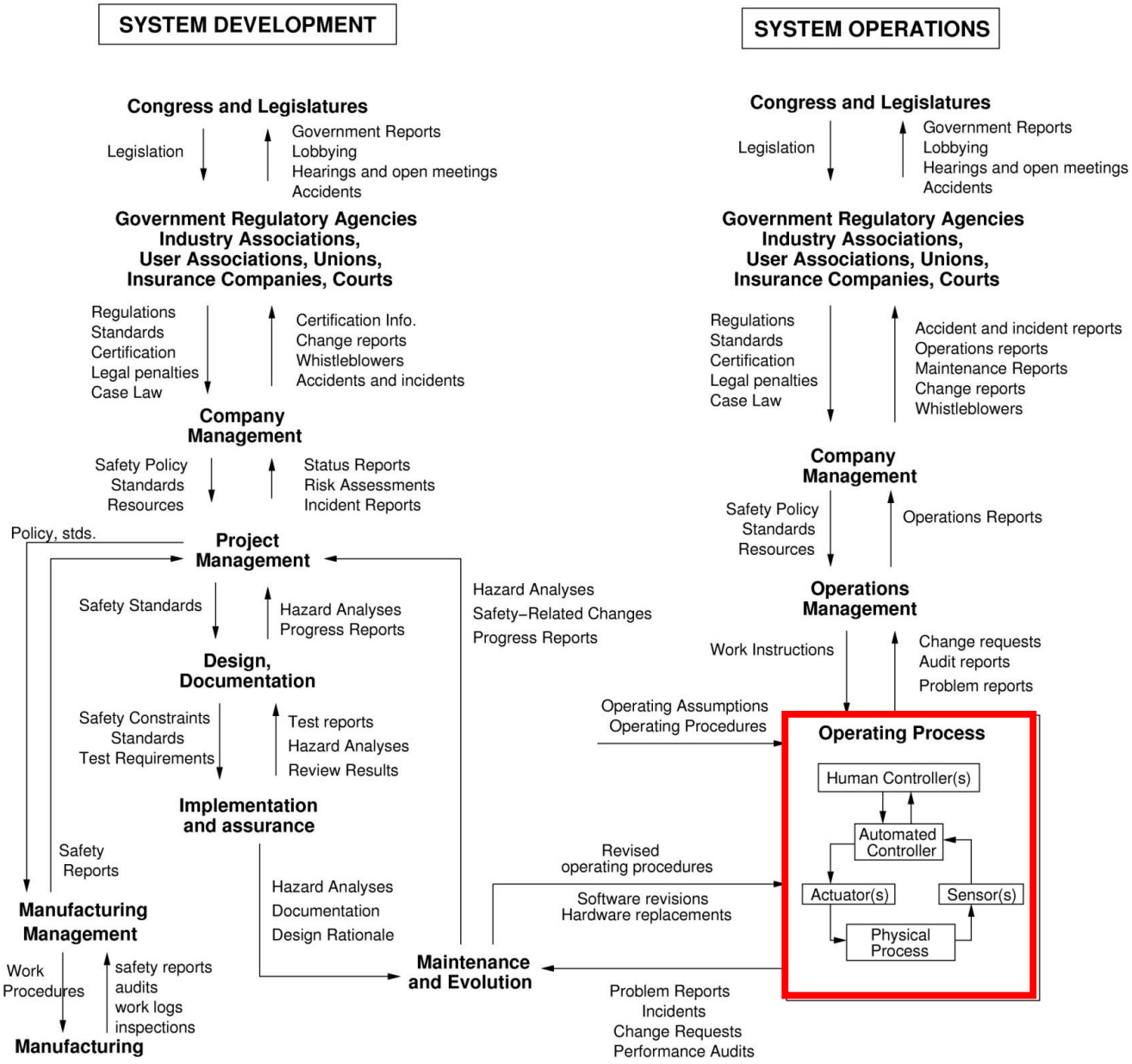
ESW p206:
U.S. pharmaceutical safety control structure

Safety Control Structure



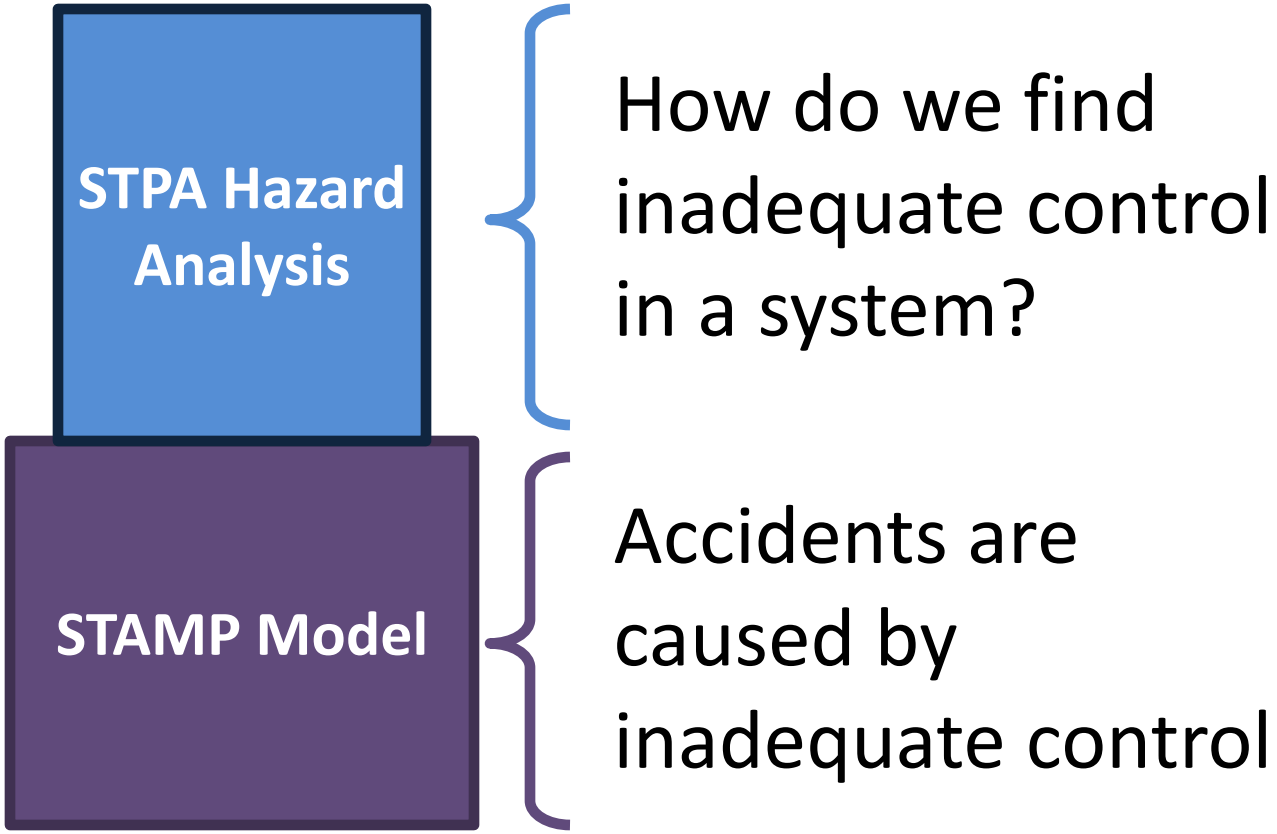
ESW p216:
Ballistic Missile
Defense System

Example Safety Control Structure



System-Theoretic Process Analysis

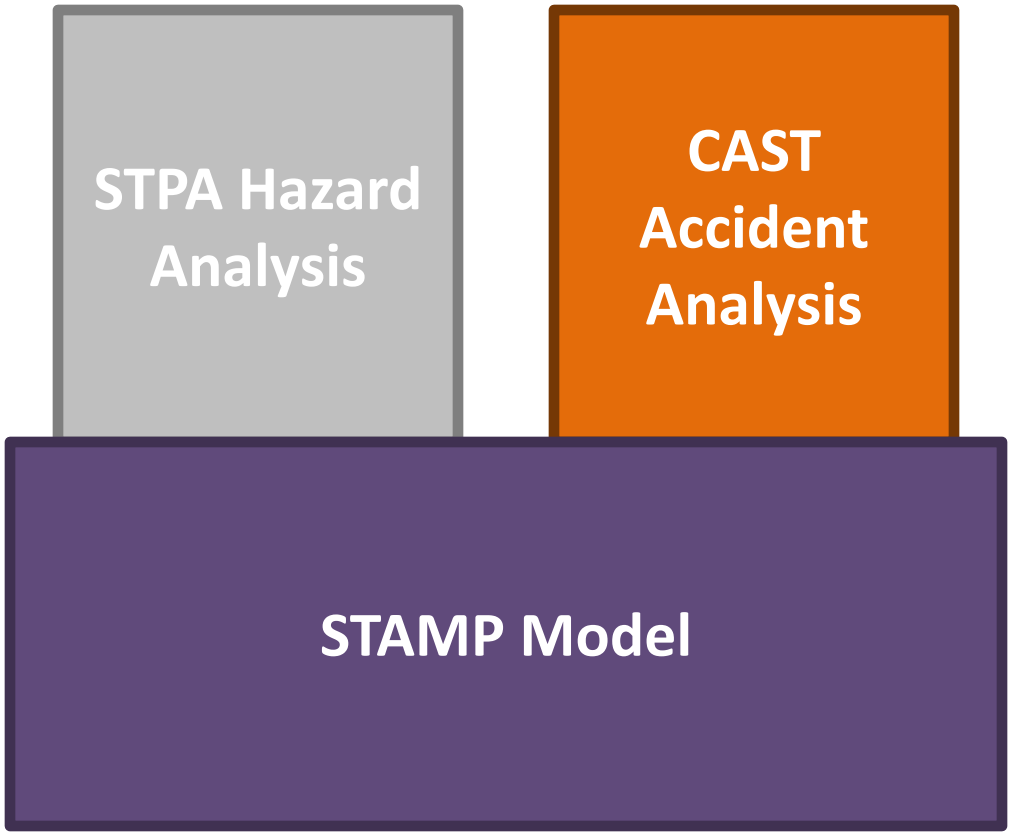
STPA



(Leveson, 2011)

Causal Analysis using System Theory

CAST



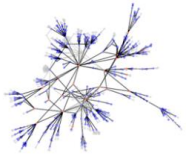
How do we find inadequate control that caused the accident?

Accidents are caused by inadequate control

(Leveson, 2011)

Advanced Tutorial

- Early Morning 9-10:30
  - STPA Hazard Analysis
  - "Guided" exercises
- Late Morning 10:45-12
  - Hands-on exercises
  - Developing S/W requirements



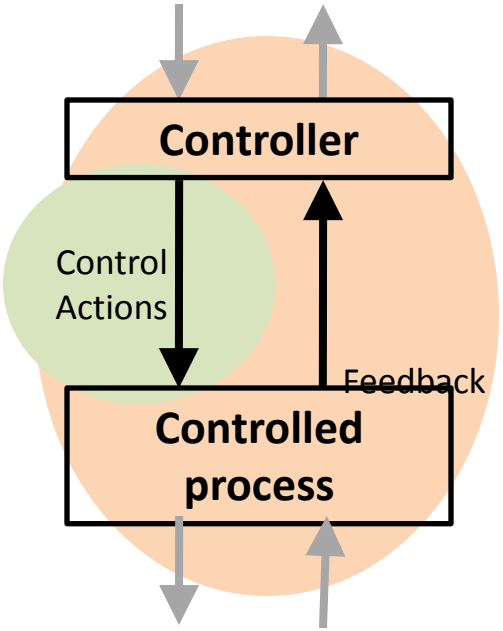
STPA Hazard Analysis

System-Theoretic Process Analysis

STPA Hazard Analysis

STAMP Model

- Identify the hazards
- Construct the control structure
- Step 1: Identify unsafe control actions
- Step 2: Identify causes of unsafe control actions



(Leveson, 2011)

Step 1: Identify Unsafe Control Actions

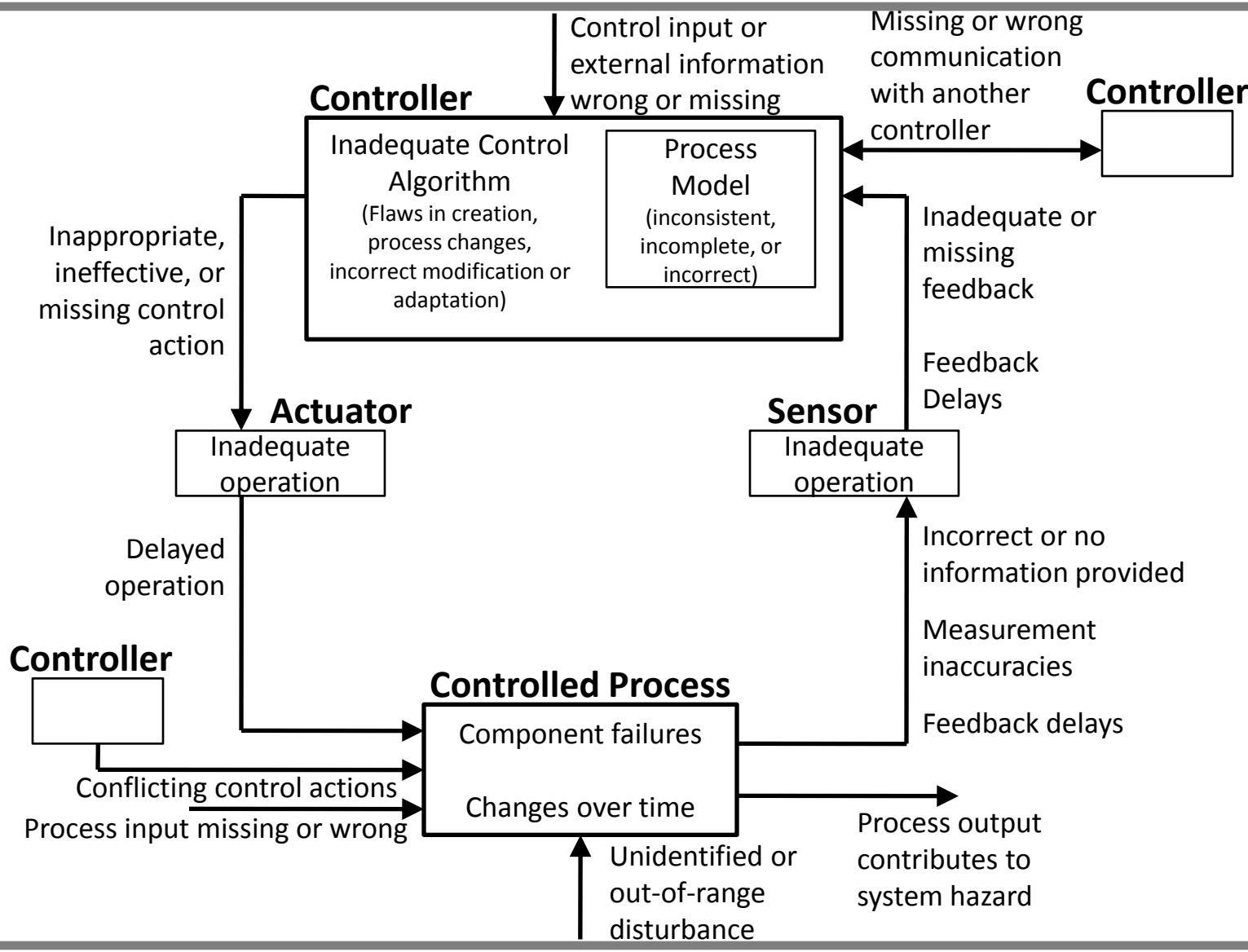
	Action required but not provided	Unsafe action provided	Incorrect Timing/ Order	Stopped Too Soon / Applied too long
Action (Role)				

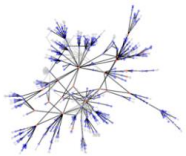
Step 1: Identify Unsafe Control Actions

(a more rigorous approach)

Control Action	Process Model Variable 1	Process Model Variable 2	Process Model Variable 3	Hazardous?

Step 2: STPA Control Flaws





Simple STPA Exercise

a new in-trail procedure
for trans-oceanic flights

Example System: Aviation

Accident (Loss): Two aircraft collide

STPA Exercise

- Identify Hazards
- Draw the control structure
  - Identify major components and controllers
  - Label the control/feedback arrows
- Identify Unsafe Control Actions (UCAs)
  - Control Table:
    Not given, Given incorrectly, Wrong timing,
    Stopped too soon
  - Create corresponding safety constraints
- Identify causal factors
  - Identify controller process models
  - Analyze controller, control path, feedback path, process

Hazard

- Definition: A system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to an accident (loss).
- Something we can ***control***
- Examples:

Accident	Hazard
Satellite becomes lost or unrecoverable	Satellite maneuvers out of orbit
People are exposed to toxic chemicals	Toxic chemicals are released into the atmosphere
People are irradiated	Nuclear power plant experiences nuclear meltdown
People are poisoned by food	Food products containing pathogens are sold

Accident (Loss): Two aircraft collide

Hazard: ?

Accident (Loss): Two aircraft collide

Hazard: two aircraft violate minimum separation

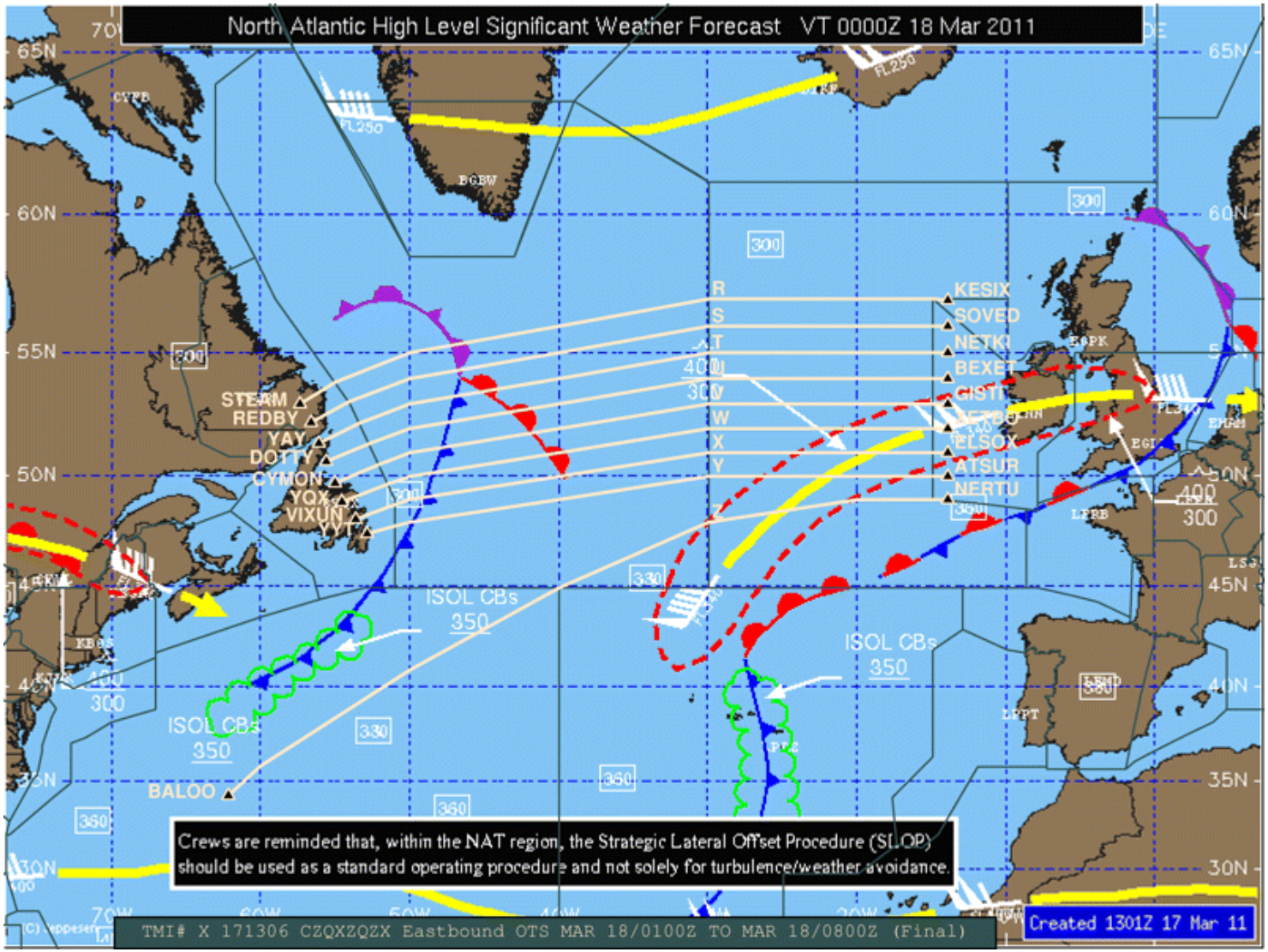
Identifying Hazards

- Loss (accident)
  - Two aircraft collide
  - Aircraft crashes into terrain / ocean
- Hazards
  - Two aircraft violate minimum separation
  - Aircraft enters unsafe atmospheric region
  - Aircraft enters uncontrolled state
  - Aircraft enters unsafe attitude
  - Aircraft enters prohibited area

STPA Exercise

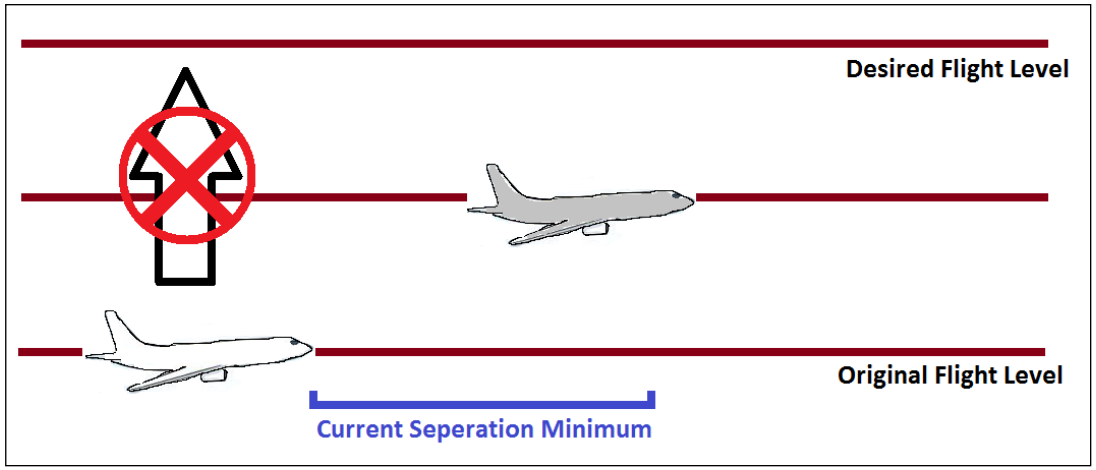
- Identify Hazards
- Draw the control structure
  - Identify major components and controllers
  - Label the control/feedback arrows
- Identify Unsafe Control Actions (UCAs)
  - Control Table:
    Not given, Given incorrectly, Wrong timing,
    Stopped too soon
  - Create corresponding safety constraints
- Identify causal factors
  - Identify controller process models
  - Analyze controller, control path, feedback path, process

North Atlantic Tracks



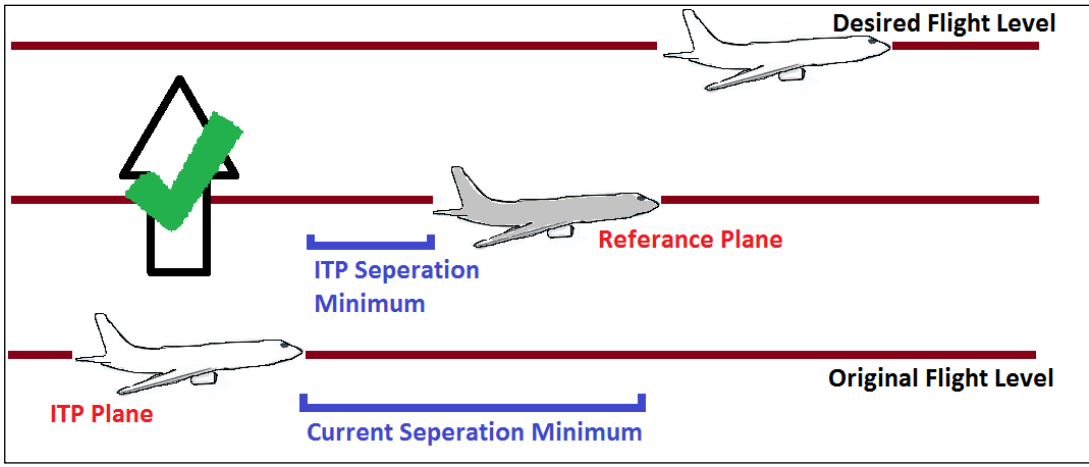
NextGen In-Trail Procedure (ITP)

Current State



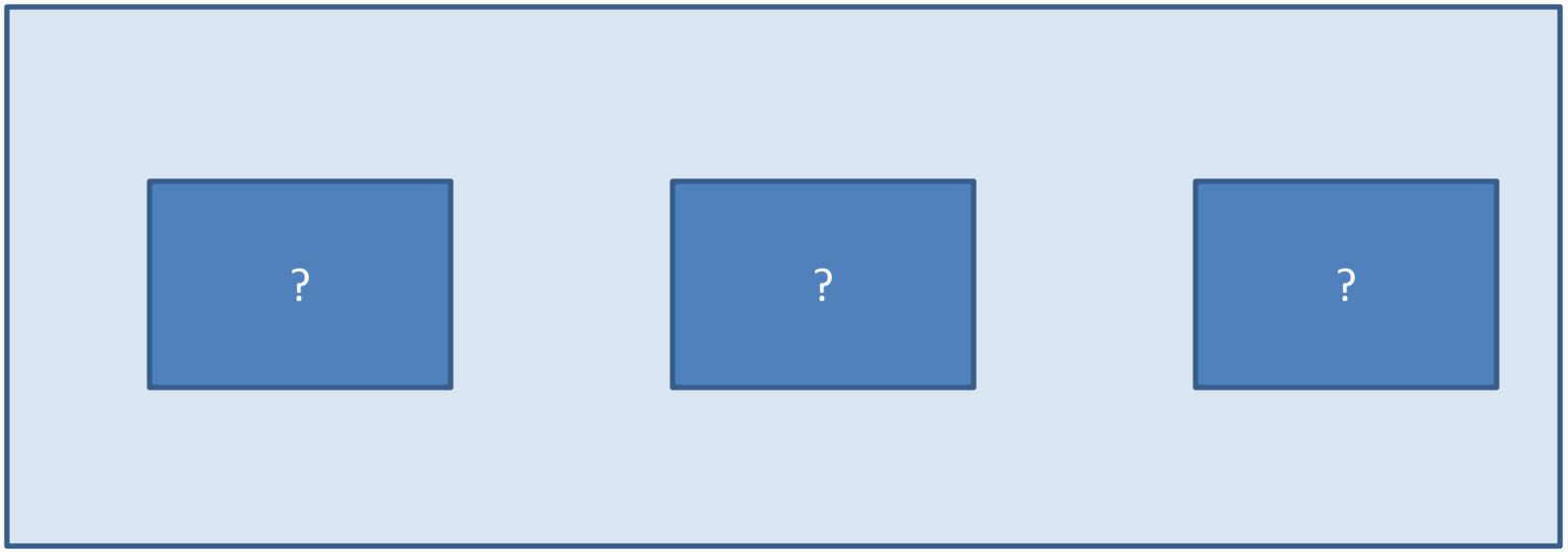
- Pilots will have separation information
- Pilots decide when to request a passing maneuver
- Air Traffic Control approves/denies request

Proposed Change



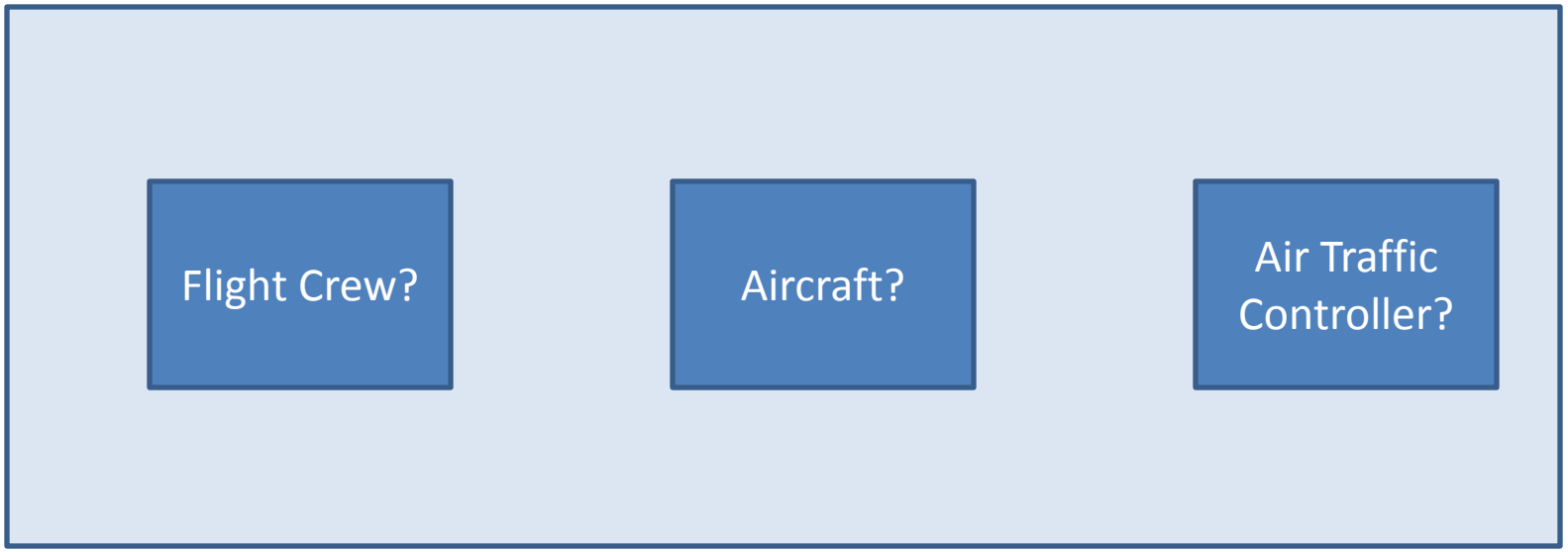
STPA Analysis

- High-level (simple) Control Structure
  - Main components and controllers?



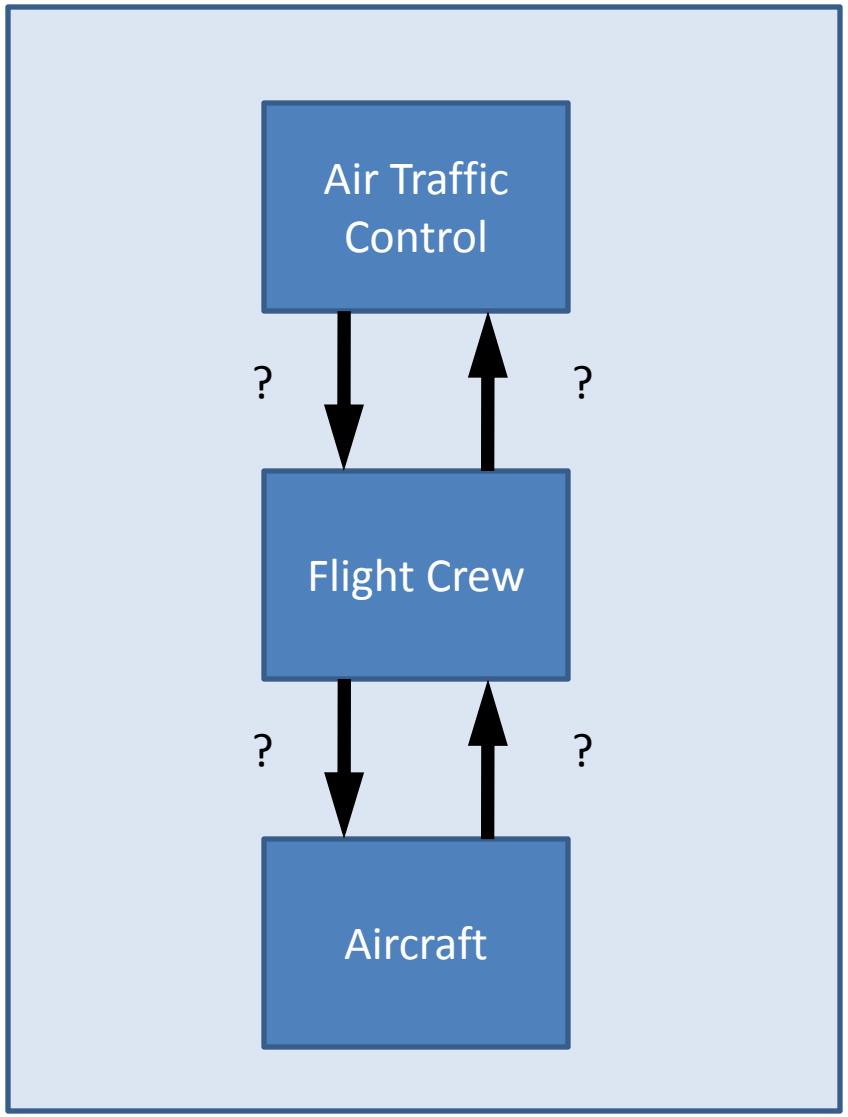
STPA Analysis

- High-level (simple) Control Structure
  - Who controls who?



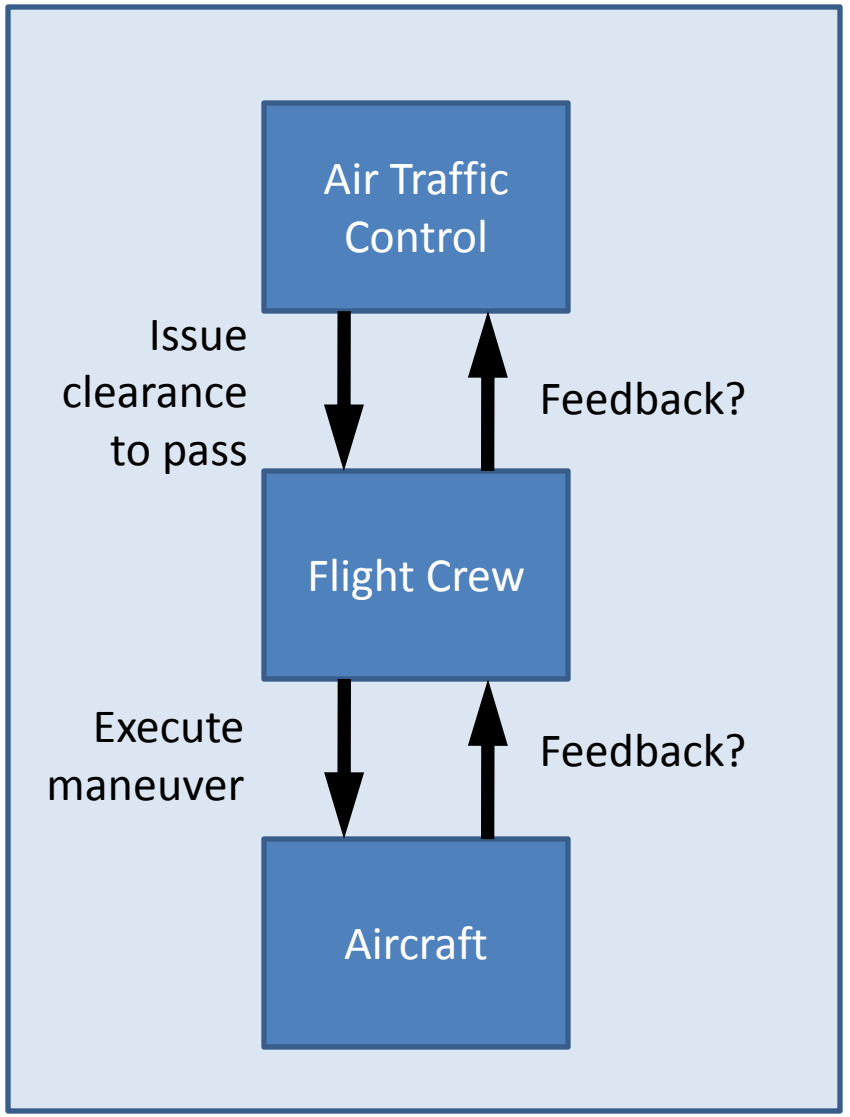
STPA Analysis

- High-level (simple) Control Structure
  - What commands are sent?



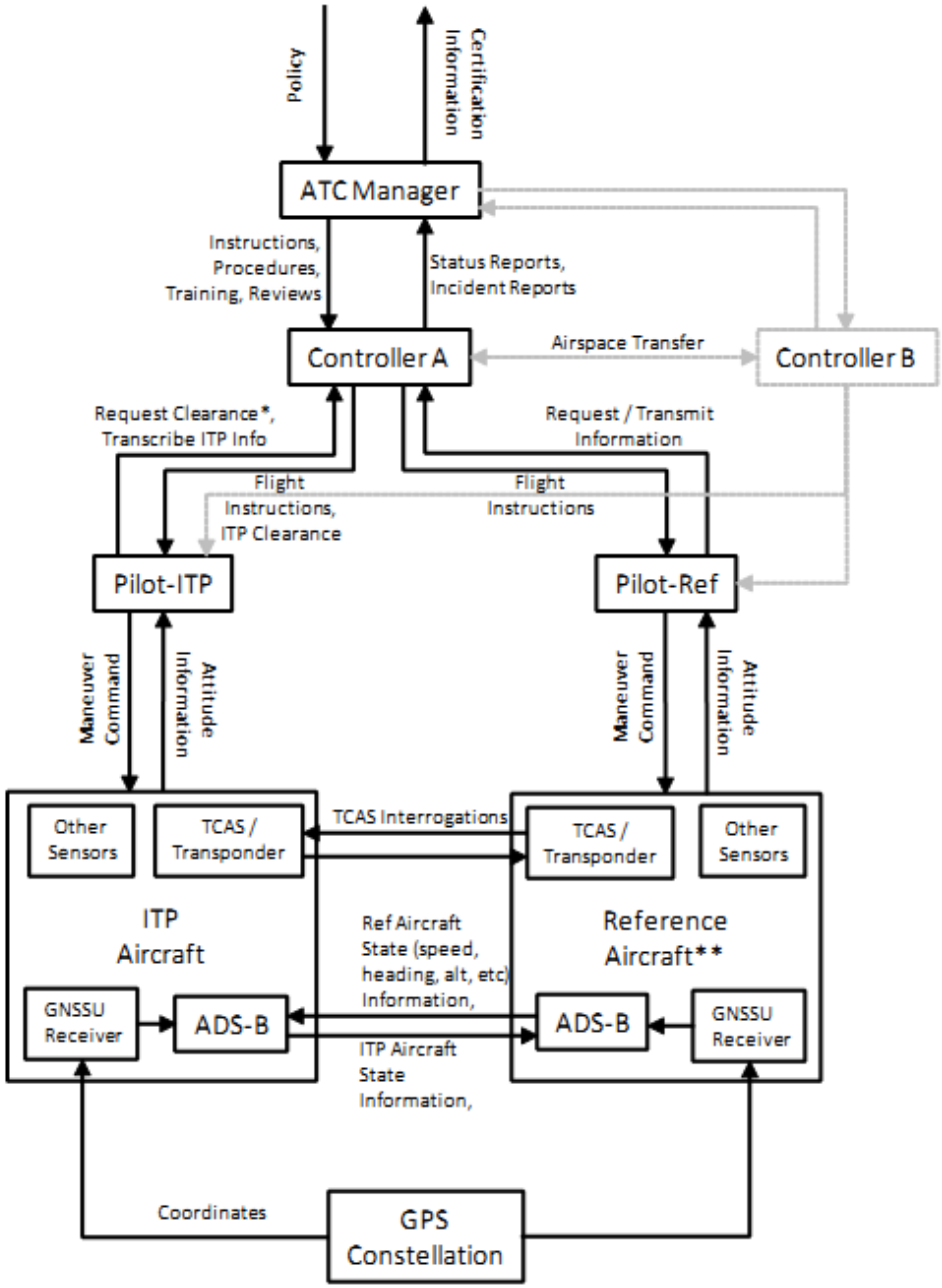
STPA Analysis

- High-level (simple) Control Structure



STPA Analysis

- More complex control structure



STPA Exercise

- Identify Hazards
- Draw the control structure
  - Identify major components and controllers
  - Label the control/feedback arrows
- Identify Unsafe Control Actions (UCAs)
  - Control Table:
    Not given, Given incorrectly, Wrong timing,
    Stopped too soon
  - Create corresponding safety constraints
- Identify causal factors
  - Identify controller process models
  - Analyze controller, control path, feedback path, process

STPA Analysis:

Identify Unsafe Control Actions

Flight Crew Action (Role)	Action required but not provided	Unsafe action provided	Incorrect Timing/ Order	Stopped Too Soon
Execute Passing Maneuver	Pilot does not execute maneuver once it is approved			

STPA Analysis:

Identify Unsafe Control Actions

Flight Crew Action (Role)	Action required but not provided	Unsafe action provided	Incorrect Timing/ Order	Stopped Too Soon
Execute passing maneuver	Pilot does not execute maneuver Aircraft remains In-Trail	Perform ITP when ITP criteria are not met or request has been refused Pilot instructs incorrect attitude, e.g. throttle and/or pitch	Crew starts maneuver late after having re-verified ITP criteria Pilot throttles before achieving necessary altitude	Crew does not complete entire maneuver e.g. Aircraft does not achieve necessary altitude or speed

STPA Analysis: Identify UCA's

Flight Crew Action (Role)	Action required but not provided	Unsafe action provided	Incorrect Timing/ Order	Stopped Too Soon
Read Back Clearance	Crew does not read-back ITP clearance	Confirm clearance but clearance had not been granted	Reads back clearance in non-standard order	
Verify ITP Criteria to Confirm Validity of Clearance	Crew does not perform ITP criteria verification	Confirm clearance when criteria are not met	Verifies criteria late after clearance was initially granted or too early before maneuver is actually performed	
Perform ITP Maneuver	Pilot does not execute maneuver Aircraft remains In-Trail	Perform ITP when ITP criteria are not met or request has been refused Pilot instructs incorrect attitude, e.g. throttle and/or pitch	Crew starts maneuver late after having re-verified ITP criteria Pilot throttles before achieving necessary altitude	Crew does not complete entire maneuver e.g. Aircraft does not achieve necessary altitude or speed
Provide data to ATC & other aircraft	Does not communicate position & attitude information	Transmit unnecessary data or information Transmit incorrect data		

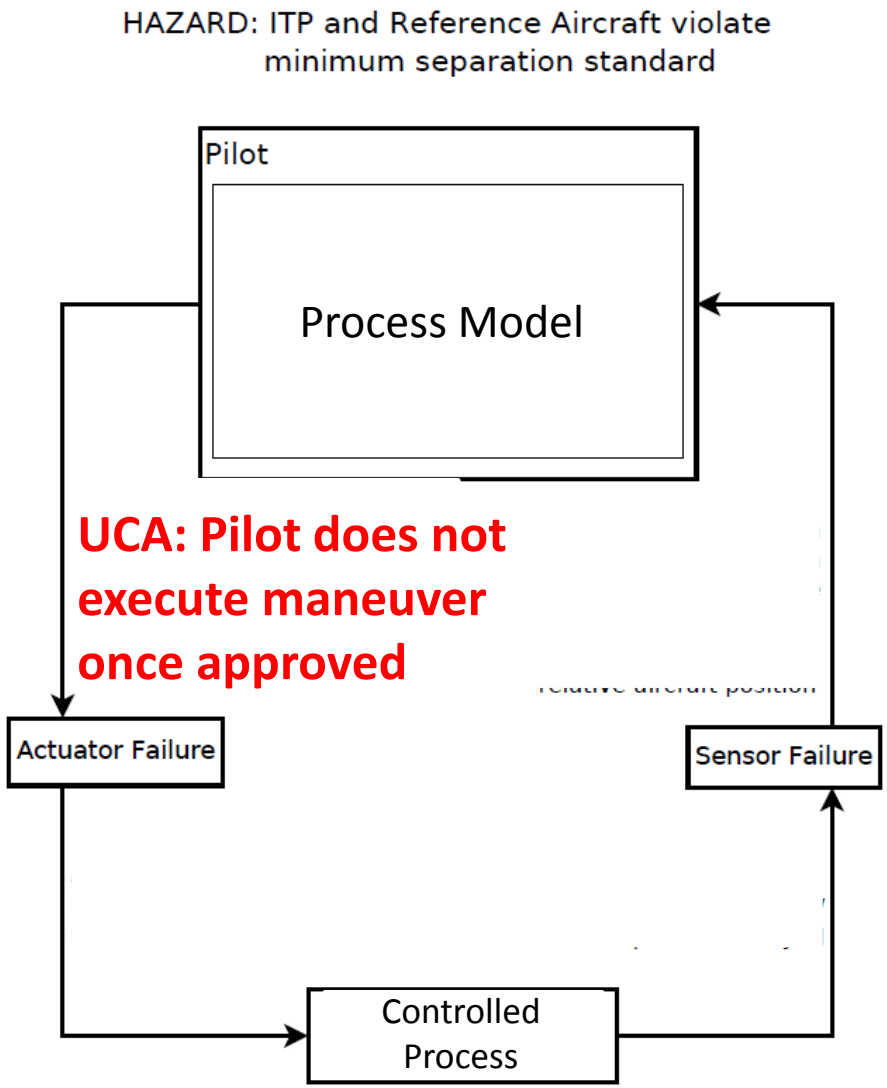
Defining Safety Constraints

Unsafe Control Action	Safety Constraint
Pilot does not execute maneuver once it is approved	Pilot must execute maneuver once it is approved
Pilot performs ITP when ITP criteria are not met or request has been refused	Pilot must not perform ITP when criteria are not met or request has been refused
Pilot starts maneuver late after having re-verified ITP criteria	Pilot must start maneuver within X minutes of re-verifying ITP criteria

STPA Exercise

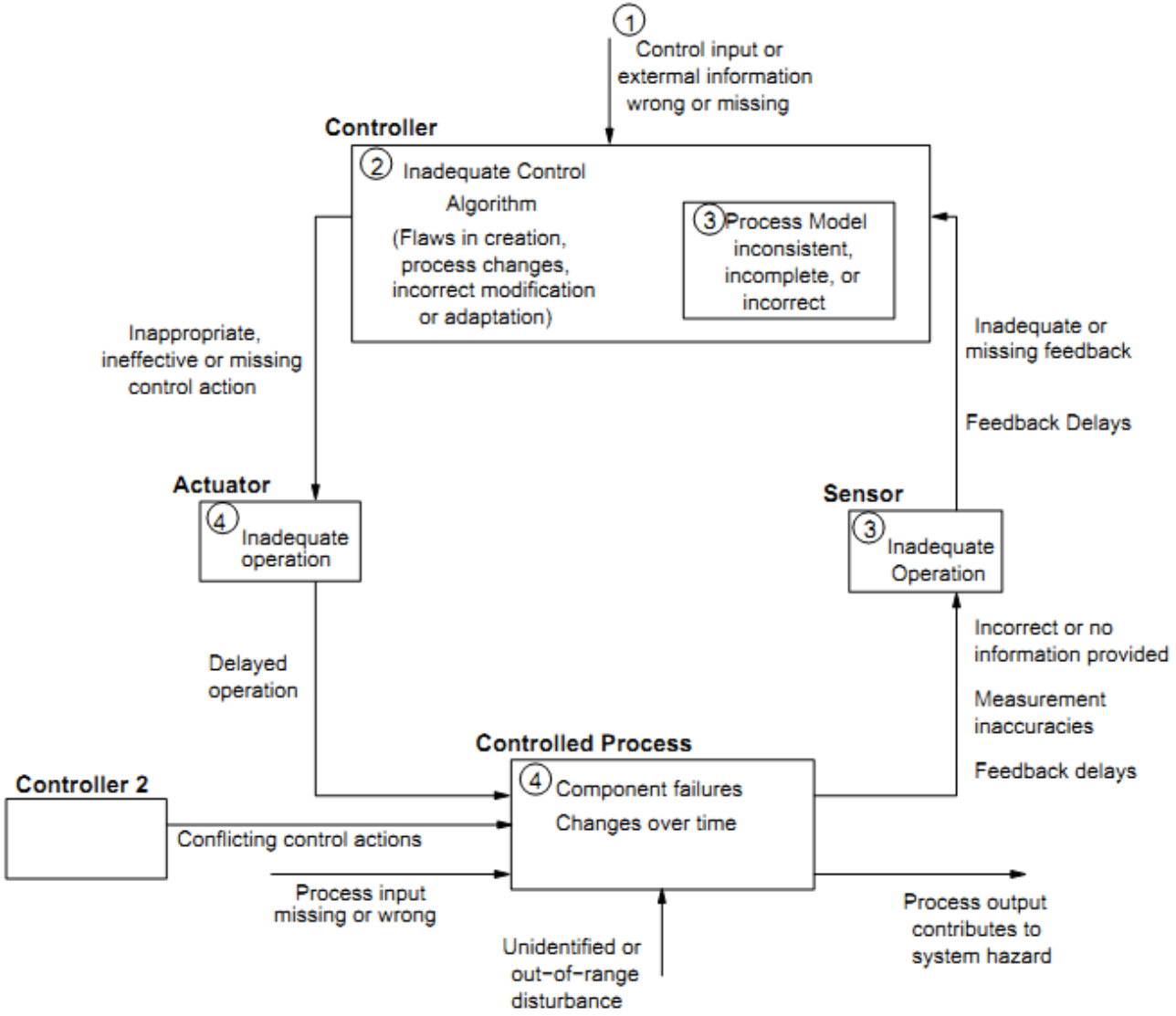
- Identify Hazards
- Draw the control structure
  - Identify major components and controllers
  - Label the control/feedback arrows
- Identify Unsafe Control Actions (UCAs)
  - Control Table:
    Not given, Given incorrectly, Wrong timing,
    Stopped too soon
  - Create corresponding safety constraints
- Identify causal factors
  - Identify controller process models
  - Analyze controller, control path, feedback path, process

STPA Analysis: Causal Factors

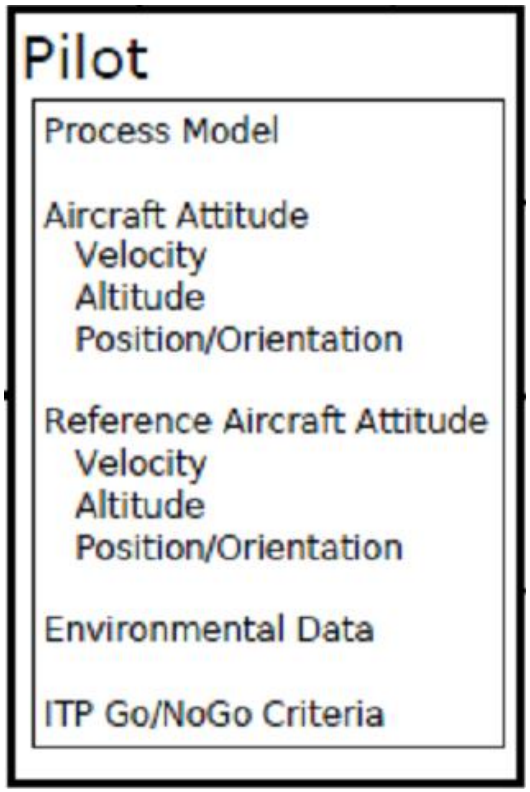


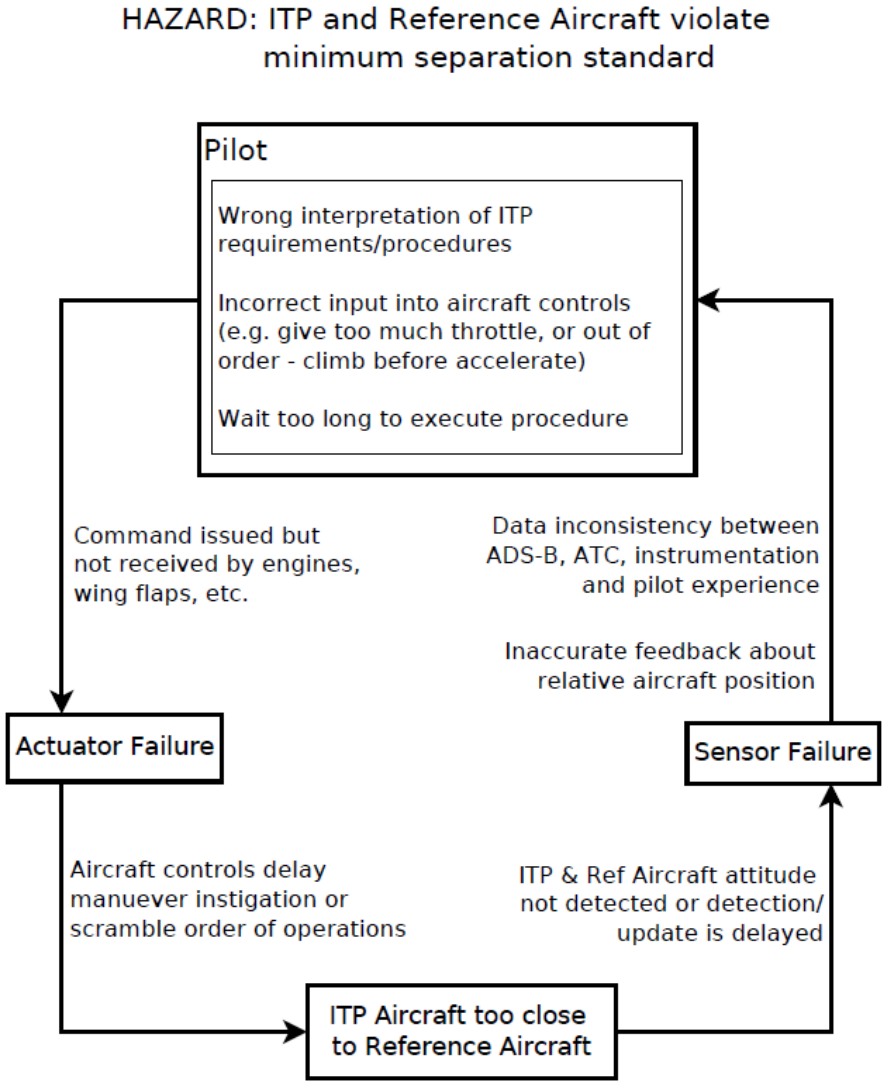
- How could this action be caused by:
  - Process model
  - Feedback
  - Sensors
  - Etc?

Hint: Causal Factors

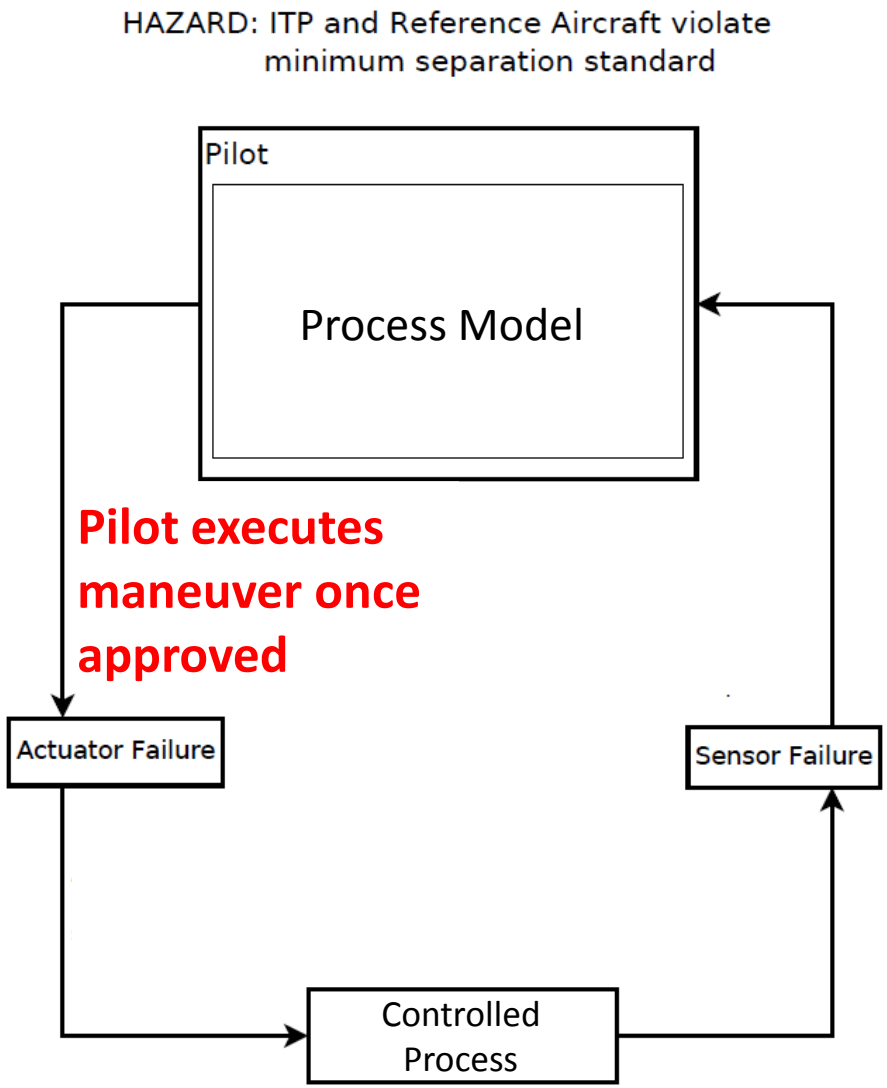


STPA Analysis: Causal Factors





STPA Analysis: Causal Factors



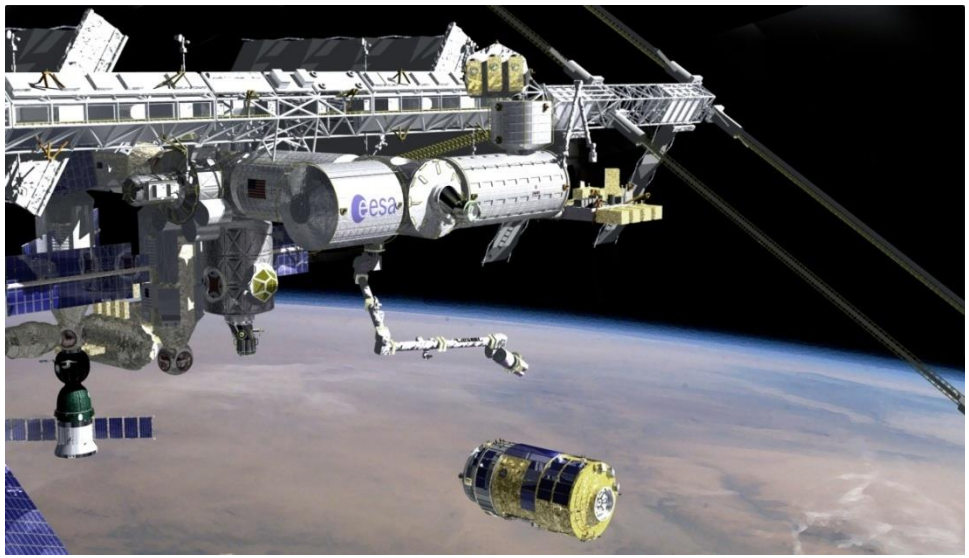
Safety Constraint: Maneuver must be executed once approved

- How else could the Safety Constraint be violated?
  - Process model
  - Feedback
  - Sensors
  - Etc?

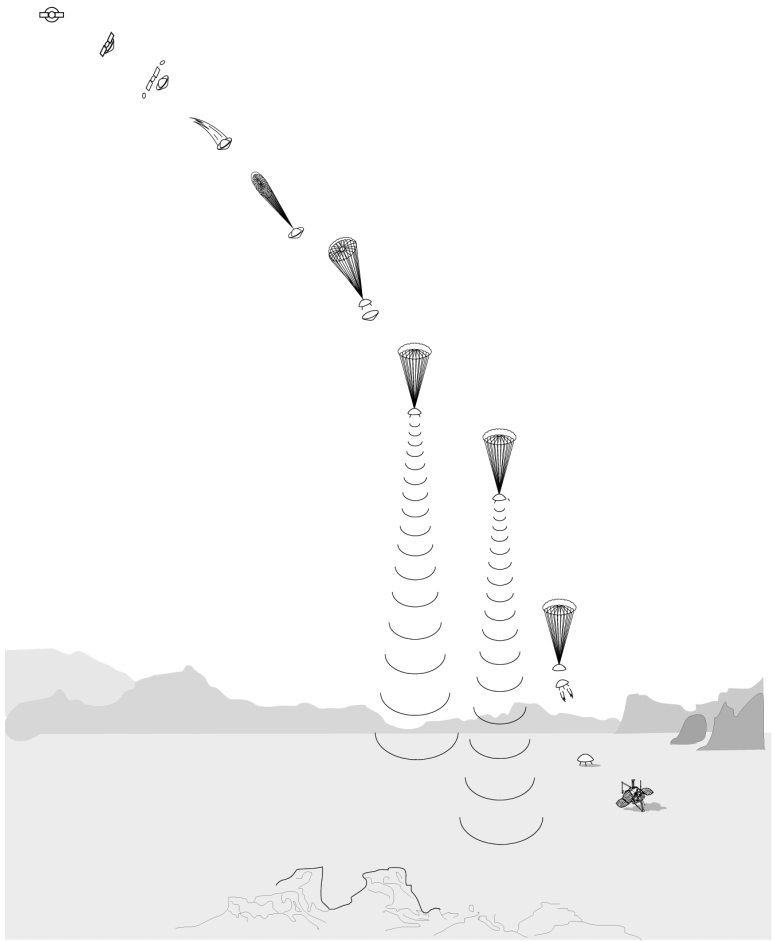
STPA Group Exercise

Choose a system to analyze:

Or pick your own!!!



International Space Station unmanned cargo vehicle



Mars Lander Descent Engine

STPA Group Exercise

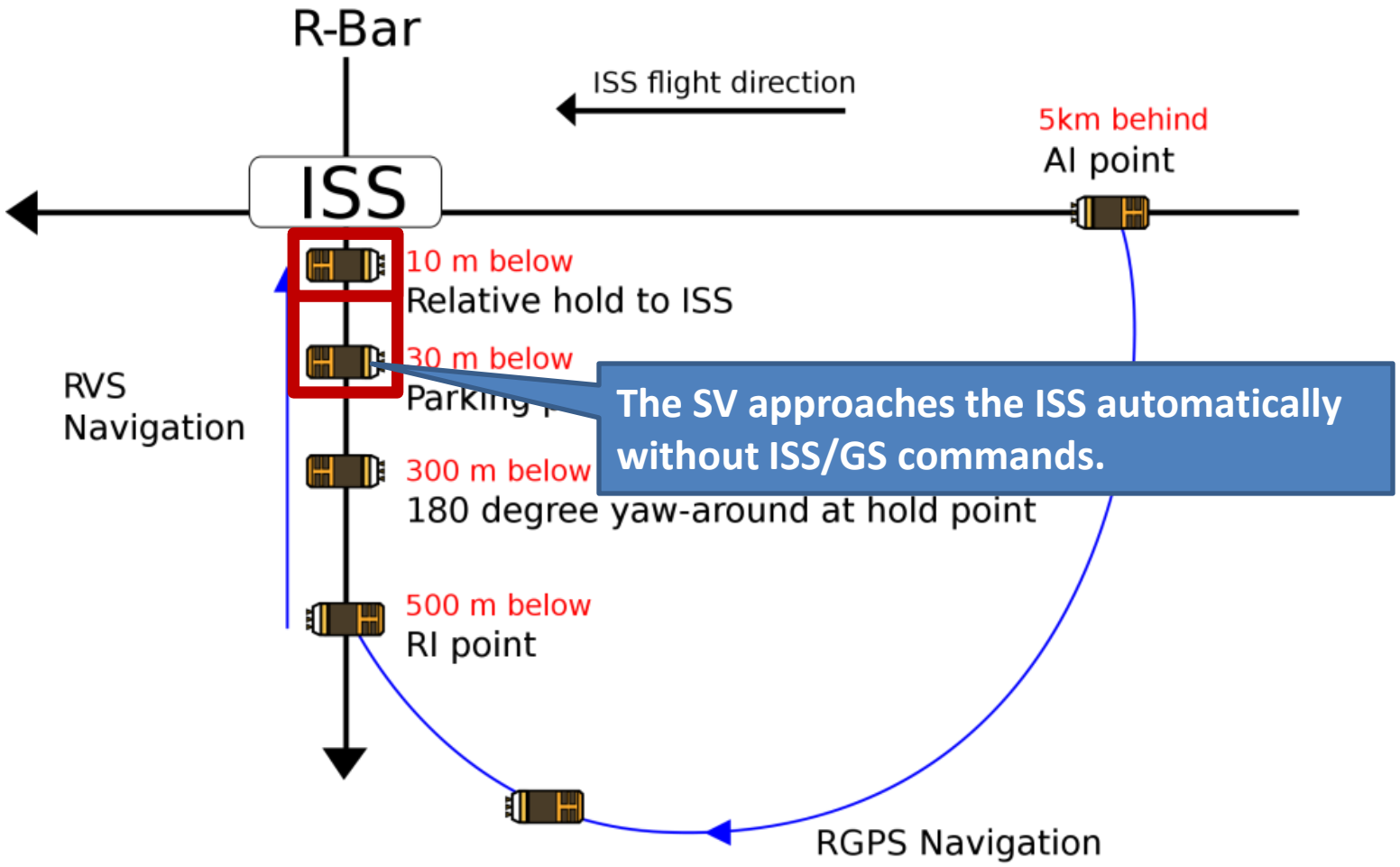
- Identify Hazards (**15 min**)
- Draw the control structure (**15 min**)
  - Identify major components and controllers
  - Label the control/feedback arrows
- Identify Unsafe Control Actions (**15 min**)
  - Control Table:
    Not given, Given incorrectly, Wrong timing, Stopped too soon
  - Create corresponding safety constraints
- Identify causal factors (**15 min**)
  - Identify controller process models
  - Analyze controller, control path, feedback path, process

Exercise – ISS Unmanned Cargo Vehicle

- Goal: deliver unmanned cargo vehicle to International Space Station

- Design constraint: must use robotic arm capture controlled by astronaut

ISS Cargo Vehicle (SV)



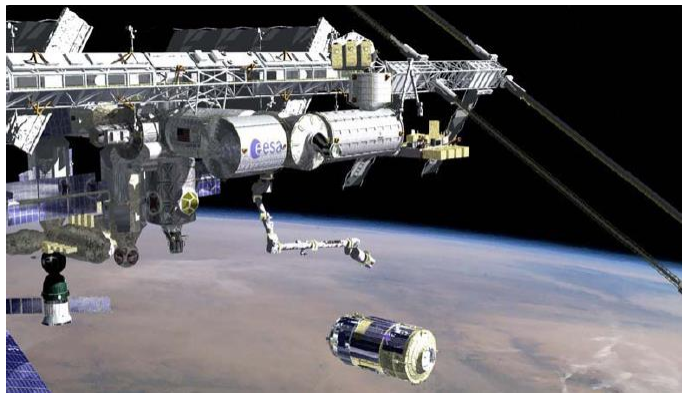
SV's approach sequence during PROX Operations

Off-Nominal Command Sequence

HTV is closer to ISS

Command	Controller	Range
HOLD	GS crew	30m – 15m
RETREAT	GS crew	15m – 10m
RETREAT	ISS crew	15m – 10m
ABORT	GS crew	10m (CP) –
ABORT	ISS crew	10m (CP) –
ABORT	HTV (Auto)	Anywhere

The most critical command is ABORT because this is the final line of defense before collision.



CP: Capture Point

➔ **If all the above commands are not provided, the HTV collides with the ISS.**

Off-Nominal Commands

TABLE. Command to avoid hazardous approach

	ABORT	RETREAT	HOLD
ISS Crew			
GS Crew			
HTV (Auto)			

Off-Nominal Commands

TABLE. Command to avoid hazardous approach

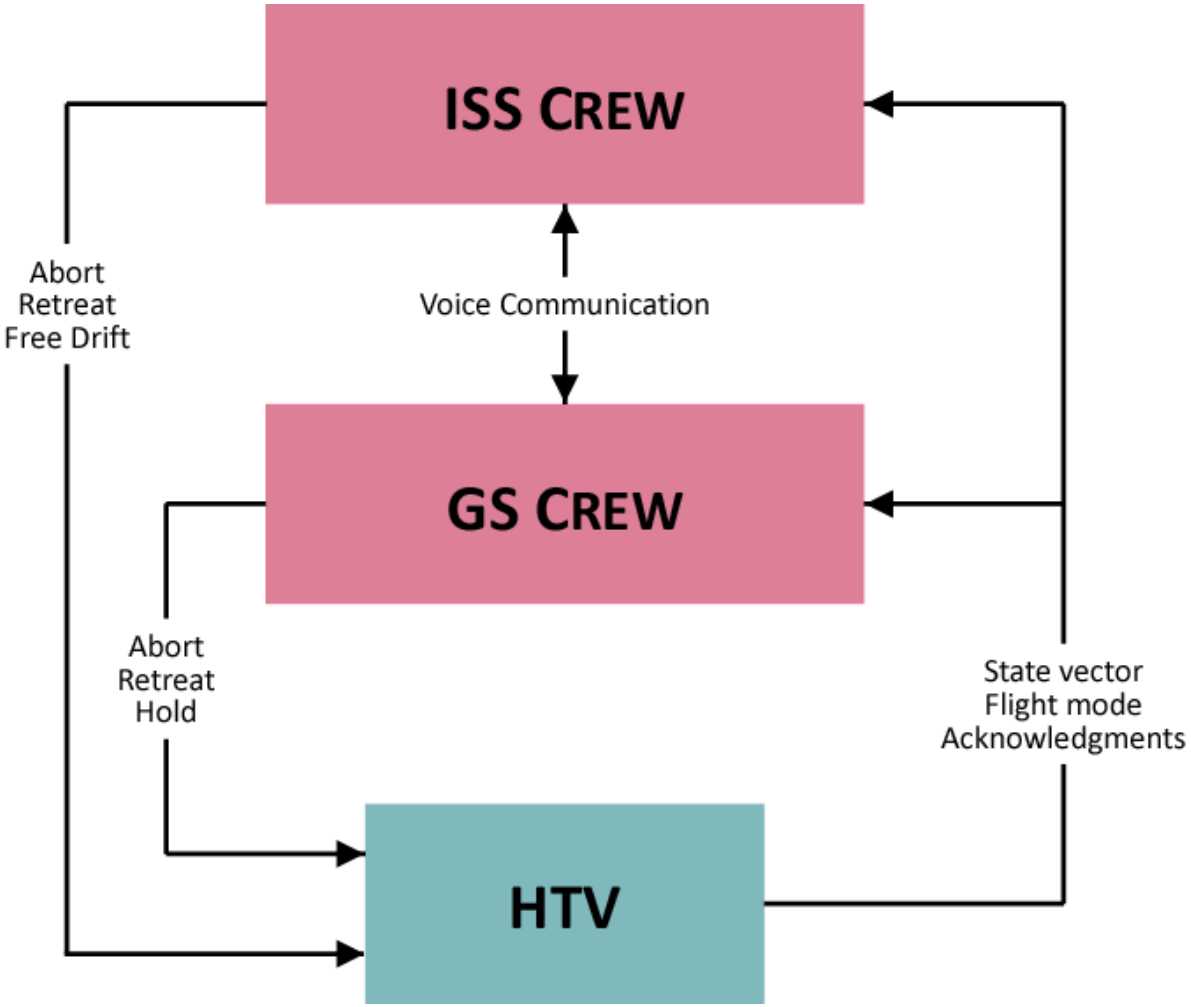
	ABORT	RETREAT	HOLD
ISS Crew	✓	✓	☑
GS Crew	✓	✓	✓
HTV (Auto)	✓	✖	✖

✓: allowed to issue (by the design/FR)
☑: not allowed but available
✖: not available (by the software design)

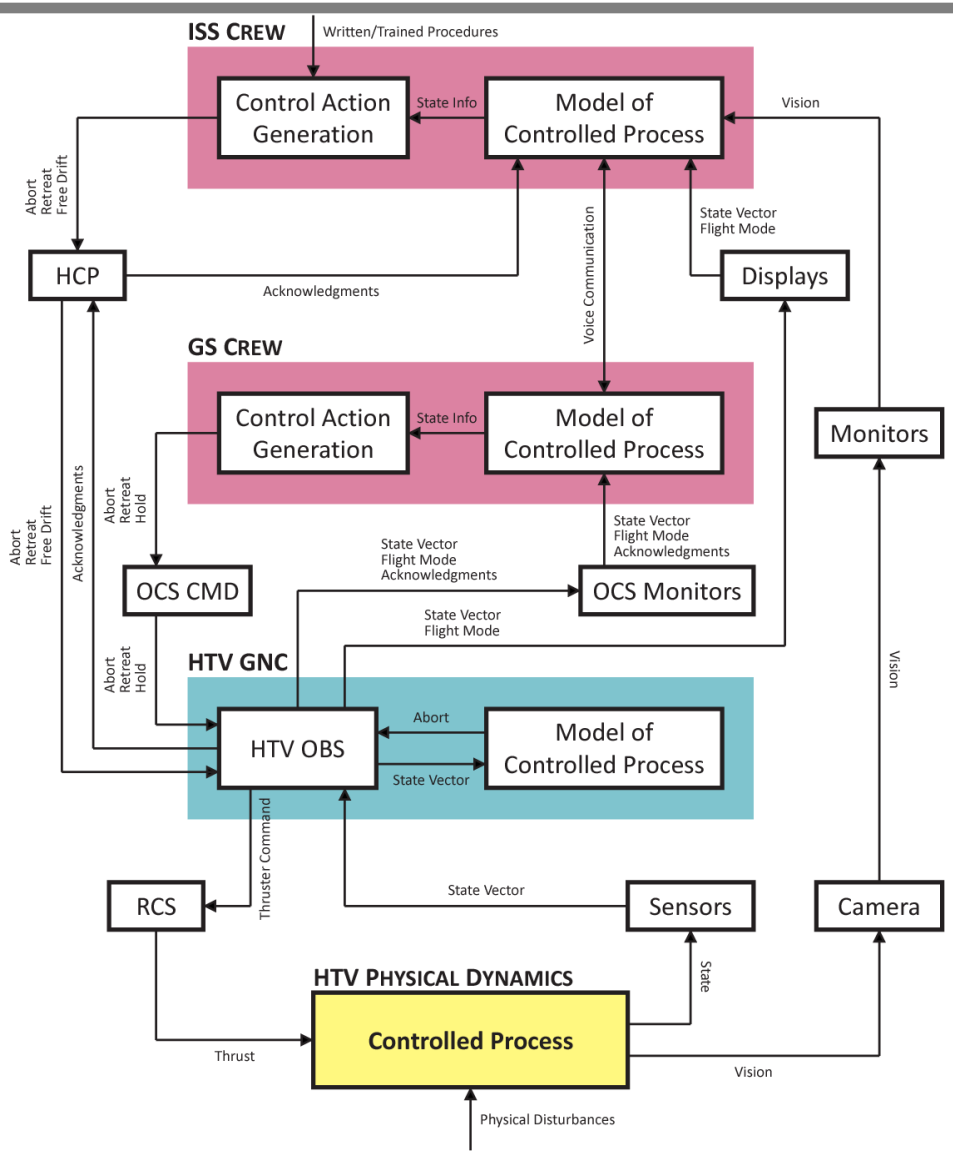
Identify Hazards

- H-1: Uncontrolled incursion into ISS
- H-2: Loss of mission (cargo not delivered to ISS)

Draw the Control Structure



Draw the Control Structure

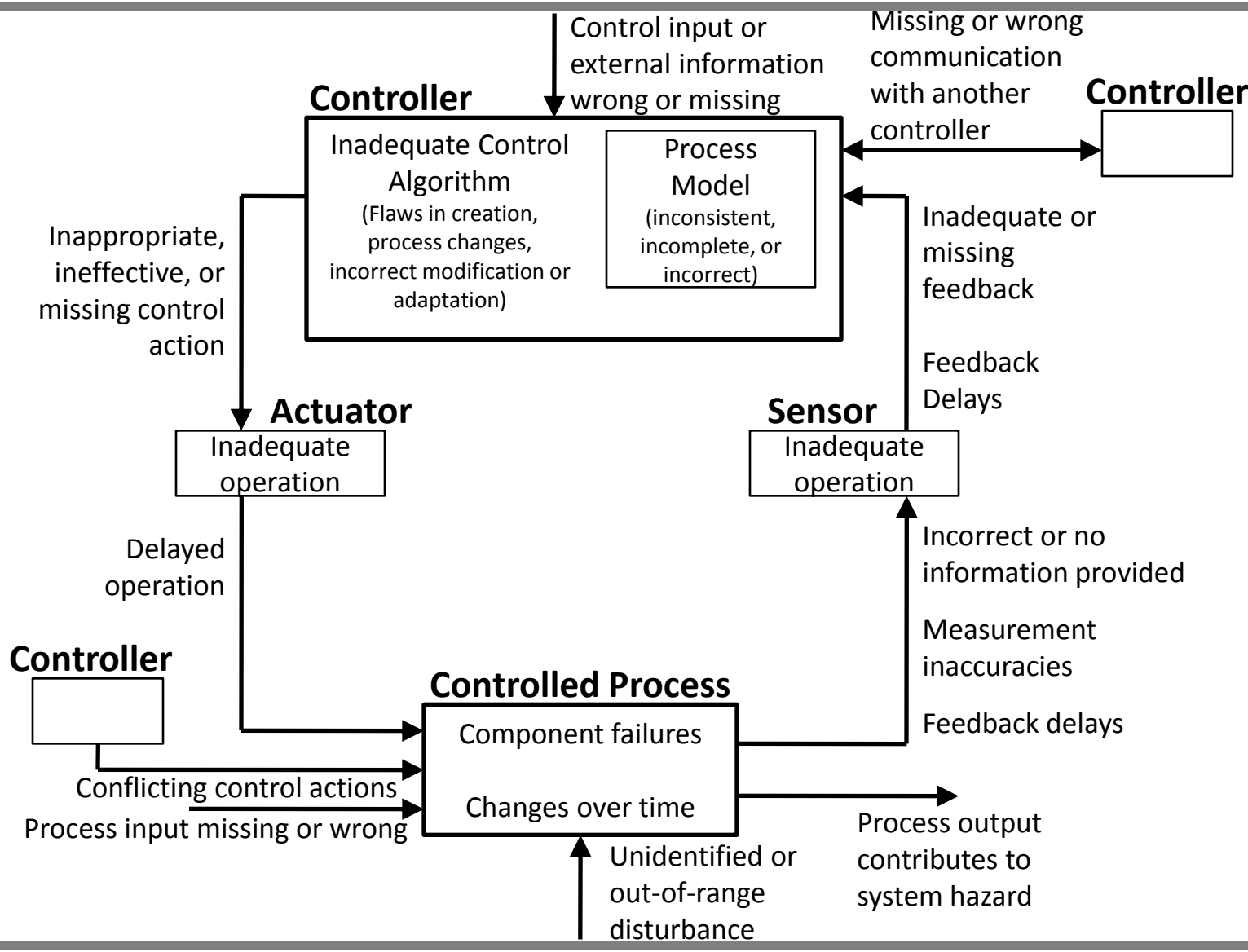


HCP: Hardware Command Panel
OCS: Operations Control System
CMD: Command
GNC: Guidance Navigation & Control
OBS: On-Board Software
RCS: Reaction Control System

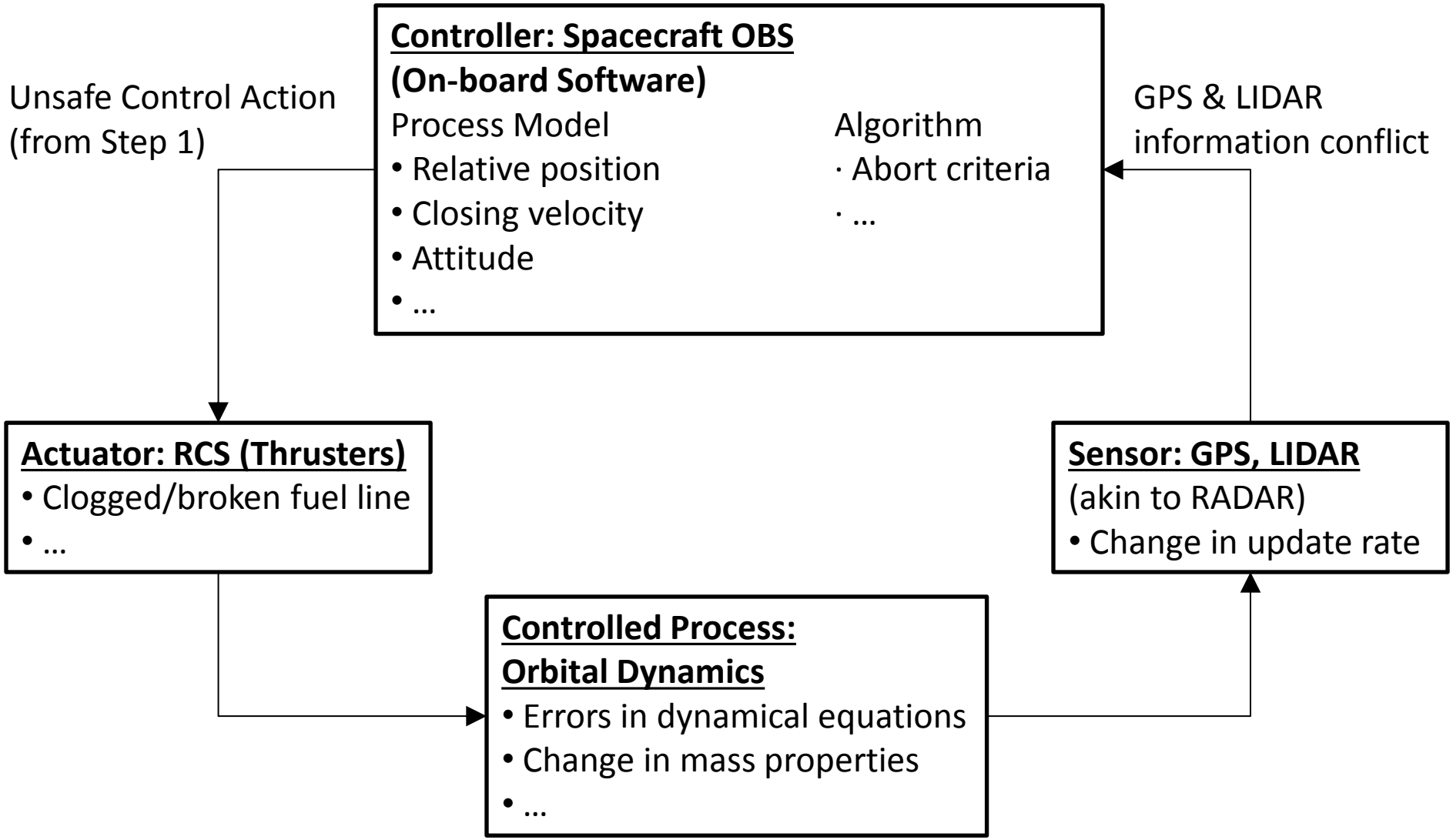
Identify Unsafe Control Actions

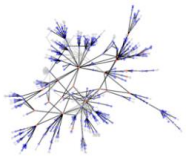
Spacecraft Software Action	Action required but not provided	Unsafe action provided	Incorrect Timing/ Order	Stopped Too Soon
Abort	Spacecraft approaches too close to ISS, or with excessive velocity [H-1] Same conditions as above, and ISS crew does not provide command	Abort provided when s/c in nominal position, velocity – [H-2] (can still potentially recover mission, but constrained by onboard fuel)	Abort provided before 'Retreat' when s/c enters warning zone [H-1]	NA (discrete command – actual abort procedure is continuous, however, and would be analyzed in another step)

Causal Factors – Remember Guidewords



Identify Causal Factors, partial example



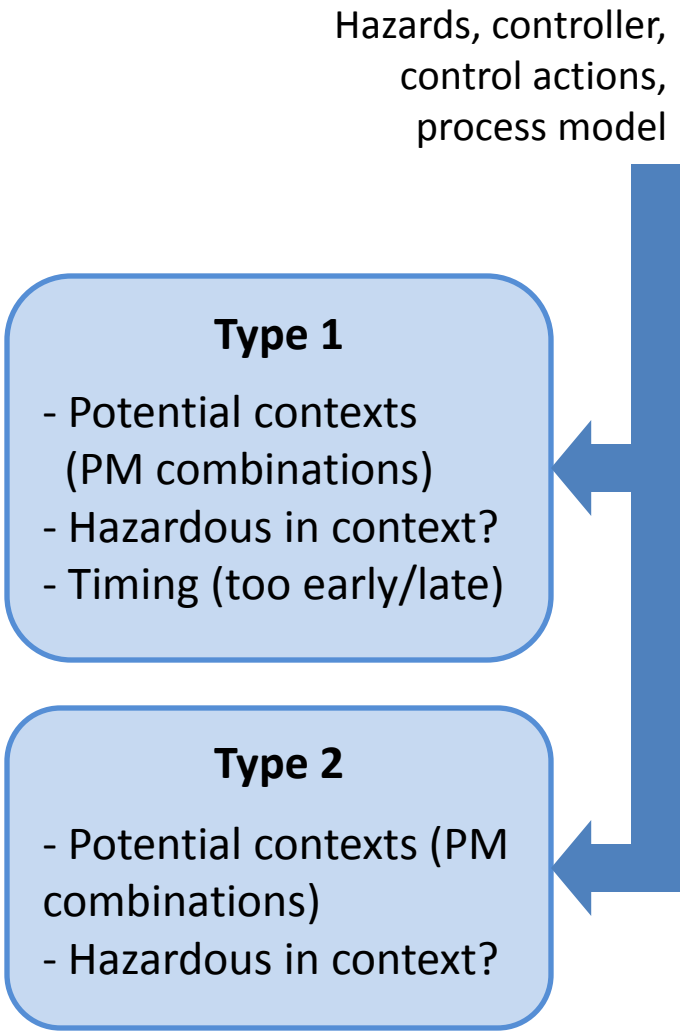


Identifying Unsafe Control Actions

(in an automatable way!)

Identifying Hazardous Control Actions

- Type 1: Providing control action causes hazard
  - 1a) Define potential contexts (combinations of process model values)
  - 1b) Determine whether the control action is hazardous in each context
  - 1c) Determine whether control action can still be hazardous if too early/too late
- Type 2: Not providing control action causes hazard
  - Same as above, but for an absence of the selected control action



Example: Train door controller

System Hazards

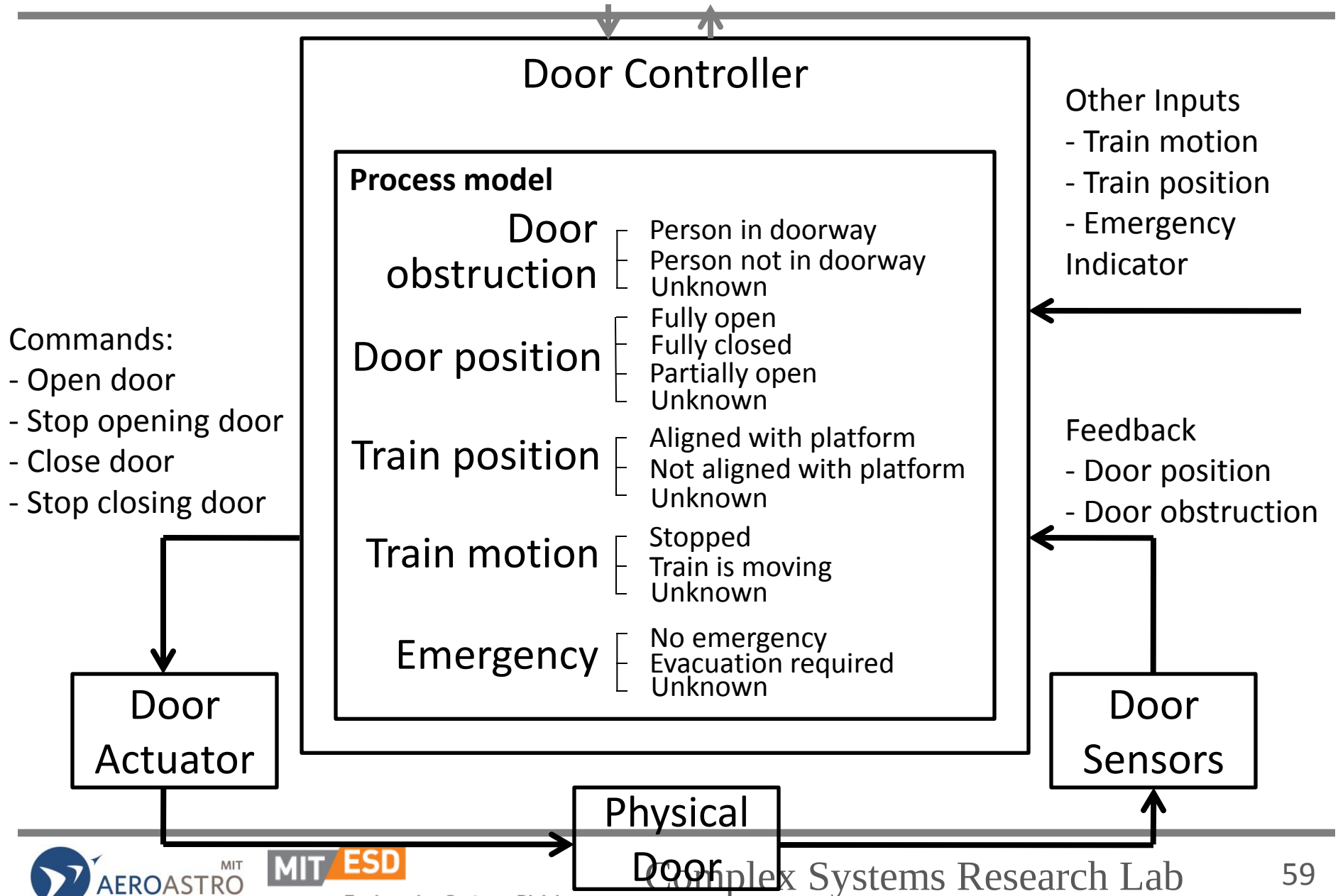
H-1: Doors close on a person in the doorway

H-2: Doors open when the train is moving or not at platform

H-3: Passengers/staff are unable to exit during an emergency

Image: http://upload.wikimedia.org/wikipedia/commons/f/fe/Mbta-redline-bombardier.jpg

Example: Control loop



1) Control action is provided

- Control action: *Door Open* command
- 1a) Define potential contexts (combinations of process model variables)

Control Action	Train Motion	Emergency	Train Position	Door Obstruction	Door Position
Door open command	Stopped	No	Aligned with platform	Not obstructed	Closed
Door open command	Stopped	No	Aligned with platform	Not obstructed	Open
Door open command	Stopped	Yes	Aligned with platform	Obstructed	Closed
…	…	…	…	…	…

1) Control action is provided

Control action: *Door Open* command

- 1a) Define potential contexts (combinations of process model variables)
- 1b) Determine whether the control action is hazardous in each context

Control Action	Train Motion	Emergency	Train Position	Door Obst. / Position	Hazardous?
Door open command	Moving	No	(doesn't matter)	(doesn't matter)	**Yes**
Door open command	Moving	Yes	(doesn't matter)	(doesn't matter)	**Yes***
Door open command	Stopped	Yes	(doesn't matter)	(doesn't matter)	**No**
Door open command	Stopped	No	Not at platform	(doesn't matter)	**Yes**
Door open command	Stopped	No	At platform	(doesn't matter)	**No**

***Design decision: In this situation, evacuate passengers to other cars. Meanwhile, stop the train and then open doors.**

1) Control action is provided

Control action: *Door Open* command

- 1a) Define potential contexts (combinations of process model variables)
- 1b) Determine whether the control action is hazardous in each context
- 1c) Determine whether control action can still be hazardous if too early/too late

Control Action	Train Motion	Emergency	Train Position	Door Obst. / Position	Hazardous ?	Hazardous if provided too early?	Hazadous if provided too late?
Door open command	Moving	No	(doesn't matter)	(doesn't matter)	**Yes**	**Yes**	**Yes**
Door open command	Moving	Yes	(doesn't matter)	(doesn't matter)	**Yes***	**Yes***	**Yes***
Door open command	Stopped	Yes	(doesn't matter)	(doesn't matter)	**No**	**No**	**Yes**
Door open command	Stopped	No	Not at platform	(doesn't matter)	**Yes**	**Yes**	**Yes**
Door open command	Stopped	No	At platform	(doesn't matter)	**No**	**No**	**No**

2) Control action is not provided

Control action: *Door Open* command

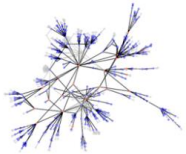
- 2a) Identify process model variables
- 2b) Determine whether the absence of control action is hazardous in each context

Control Action	Train Motion	Emergency	Train Position	Door Obst. / Pos.	Hazardous?
Door open command not provided	Stopped	Yes	(doesn't matter)	(doesn't matter)	**Yes**
Door open command not provided	Stopped	(doesn't matter)	(doesn't matter)	Closing on obstruction	**Yes**
Door open command not provided	(all others)				**No**

Resulting List of Hazardous Control Actions

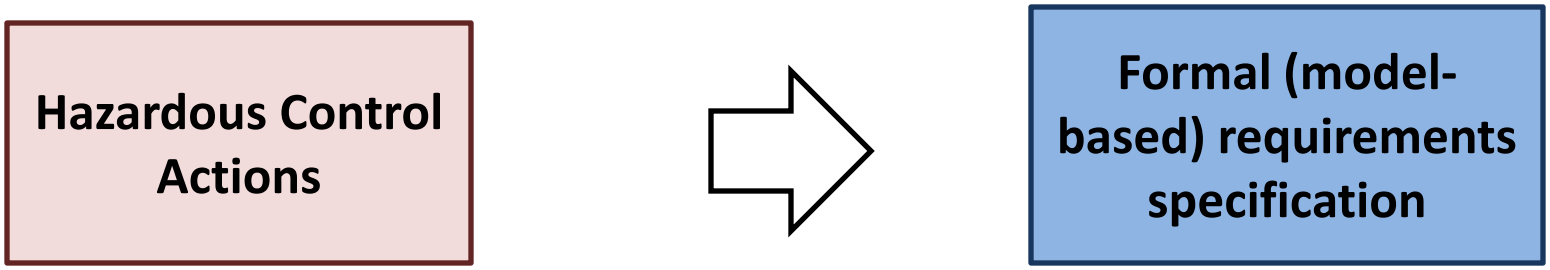
Hazardous Control Actions
Door open command provided while train is moving and there is no emergency
Door open command provided too late while train is stopped and emergency exists
Door open command provided while train is stopped, no emergency, and not at platform
Door open command provided while train is moving and emergency exists
Door open command not provided while train is stopped and emergency exists
Door open command not provided while doors are closing on someone

Parts of this can be automated!



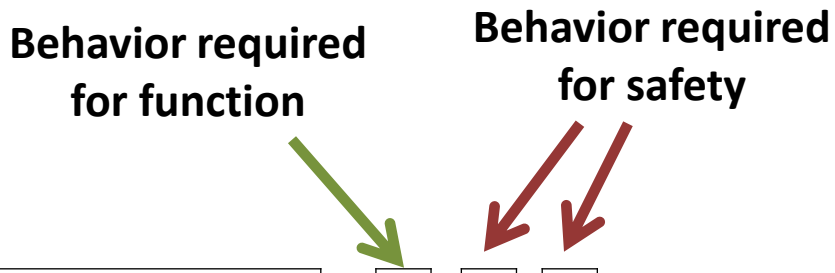
Automatically generating safety requirements

Generating safety requirements



Generating safety requirements

- Example: Generated black-box model for door controller



Provide 'Open Doors' command

Door State =	Doors not closing on person			
	Doors closing on person			T
Train Position =	Aligned with platform	T		
	Not aligned with platform			
Train Motion =	Stopped	T	T	T
	Train is moving			
Emergency =	No emergency			
	Emergency exists		T	

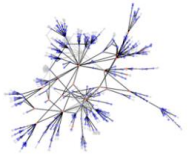
Method can help integrate safety requirements with functional requirements

Contributions

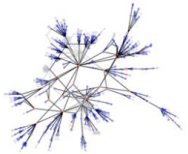
- Provides a structured way to assist in generating system requirements and safety constraints
- Can formally show consistency and internal completeness of safety-related requirements
  - Potentially could lead to formal ways to validate the safety of requirements or even help to generate requirements. No way to do this today

Impact so far

- Published in ISSC 2011
- JAXA is evaluating it on real spacecraft
- Informal presentation to Ford Research (systems engineering group); joint proposal to support this work
- Being used in MIT CSRL on air traffic control upgrades, ISS cargo vehicle, proton therapy machine, weather satellite
- NASA SBIR being written to add this functionality to a commercial requirements management toolset



Real-World Examples



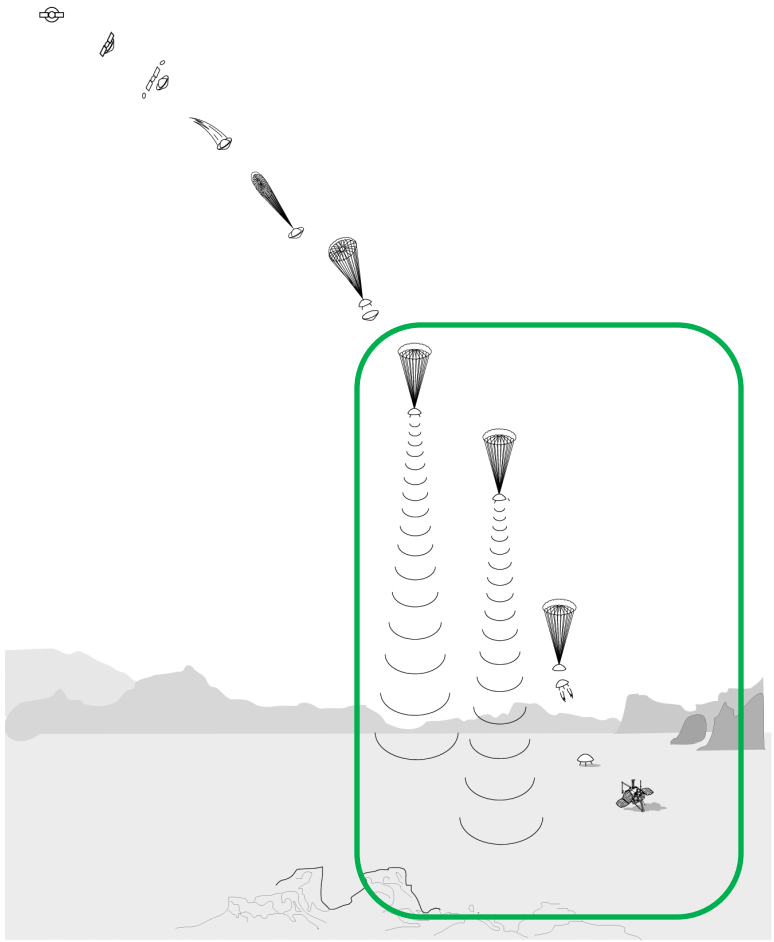
Weather Satellite

Control Action	Satellite Mode	Radar Power	Radar Mode	Hazardous Control Action?		
				Any time in this context	If provided too early	If provided too late
DPR Mode Select 'Oper-ational' *Provided*	Launch	(doesn't matter)	(doesn't matter)	H-1, H-2	H-1, H-2	H-1, H-2
	Rate Null	(doesn't matter)	(doesn't matter)	H-2	H-2	H-2
	Sun Point	(doesn't matter)	(doesn't matter)	H-2	H-2	H-2
	Δ-H	(doesn't matter)	(doesn't matter)	H-2	H-2	H-2
	Mission	On	(doesn't matter)	No	No	No
	Slew	On	(doesn't matter)	No	No	No
	Δ-V	On	(doesn't matter)	No	No	No
	Mission	Off	(doesn't matter)	H-2	H-2	H-2
	Slew	Off	(doesn't matter)	H-2	H-2	H-2
	Δ-V	Off	(doesn't matter)	H-2	H-2	H-2

Control Action	Satellite Mode	Radar Power	Radar Mode	Hazardous Control Action?
DPR Mode Select 'Safety' *Not* Provided	(doesn't matter)	(doesn't matter)	Safety	No
	(doesn't matter)	Off	Operational	H-2
	Launch	On	Operational	H-2
	Rate Null	On	Operational	H-2
	Sun Point	On	Operational	H-2
	Mission	On	Operational	No
	Slew	On	Operational	No
	Δ-V	On	Operational	No
	Δ-H	On	Operational	H-2

Exercise – Mars Lander Descent Eng.

- Goal: after arriving to Mars, hitting atmosphere, and releasing heat shield, descend from
  - 30 m/s at altitude of 500m ⇓
  - <1 m/s at ground
  - parachute stowed, engine off (this may not be realistic)
- Design constraint: must use parachute and descent engines



Hazards?

Control Actions

- Deploy parachute
- Release (cut off) parachute
- Extend landing legs
- Activate descent engines
- De-activate descent engines

→ Focus on requirements for Engine Activation Sequence

Context

- Altitude
  - > 30m
  - < 30m
  - …
- Velocity
  - > 2 m/s
  - > 1 m/s
  - < 1 m/s
  - …
- Engine
  - On
  - Off
- Parachute
  - Stowed
  - Deployed
  - Cut
- Landing legs
  - Stowed
  - Deployed

Control Actions

- "Activate Engine"

Control Action	Altitude	Velocity	Engine	Parachute	Hazardous?	Hazardous if provided too early?	Hazadous if provided too late?
Activate Engine							

Control Actions

- “Activate Engine”

Control Actions

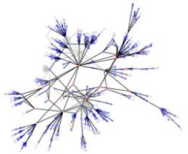
- "Activate Engine"

Control Action	Altitude	Velocity	Engine	Parachute	Hazardous?	Hazardous if provided too early?	Hazadous if provided too late?
Activate Engine							

Control Actions

- "Activate Engine"

Control Action	Altitude	Velocity	Engine	Parachute	Hazardous?	Hazardous if provided too early?	Hazadous if provided too late?
Activate Engine	> 30m						



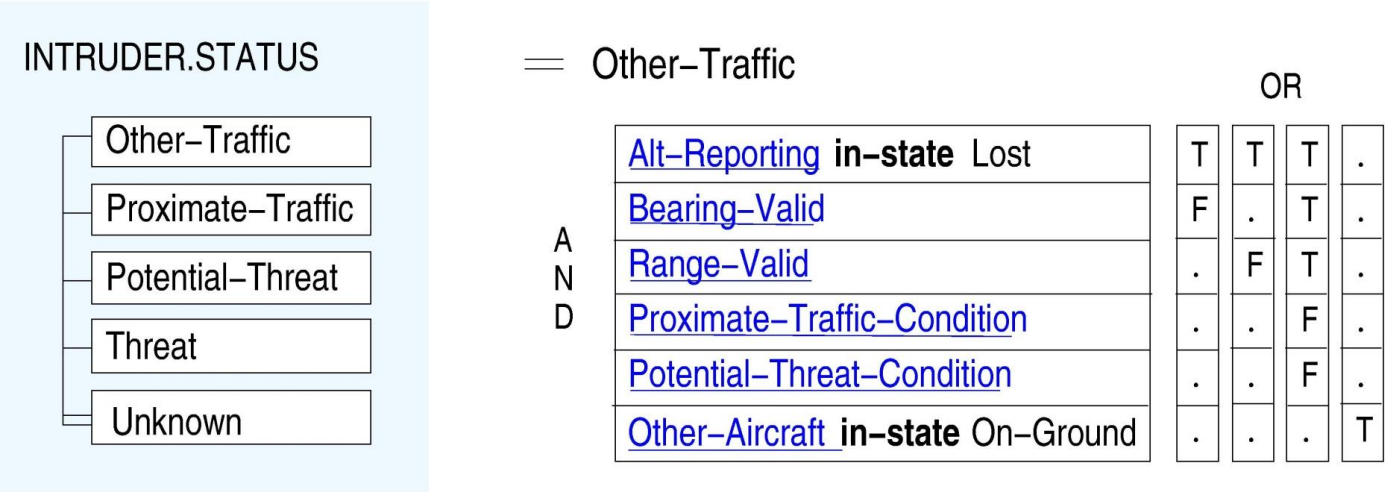
Backup

STPA Analysis: Analyze Controls

Action (Role)	Action required but not provided	Unsafe action provided	Incorrect Timing/ Order	Stopped Too Soon
Request ITP	Does not request ITP but intends to perform it (ATC & Ref Aircraft do not know of potential ITP)	Request ITP when criteria are not met Non-standard terminology leading to confusion about request	Request before criteria are met, or too late after criteria verification occurred	
Read Back Clearance	Crew does not read-back ITP clearance	Confirm clearance but clearance had not been granted	Reads back clearance in non-standard order	
Verify ITP Criteria to Confirm Validity of Clearance	Crew does not perform ITP criteria verification	Confirm clearance when criteria are not met	Verifies criteria late after clearance was initially granted or too early before maneuver is actually performed	
Perform ITP Maneuver	Pilot does not execute maneuver Aircraft remains In-Trail	Perform ITP when ITP criteria are not met or request has been refused Pilot instructs incorrect attitude, e.g. throttle and/or pitch	Crew starts maneuver late after having re-verified ITP critera Pilot throttles before achieving necessary altitude	Crew does not complete entire maneuver e.g. Aircraft does not achieve necessary altitude or speed
Provide data to ATC & other aircraft	Does not communicate position & attitude information	Transmit unnecessary data or information Transmit incorrect data		

Formal (model-based) requirements specification language

Example: SpecTRM-RL Model of TCAS II Collision Avoidance Logic

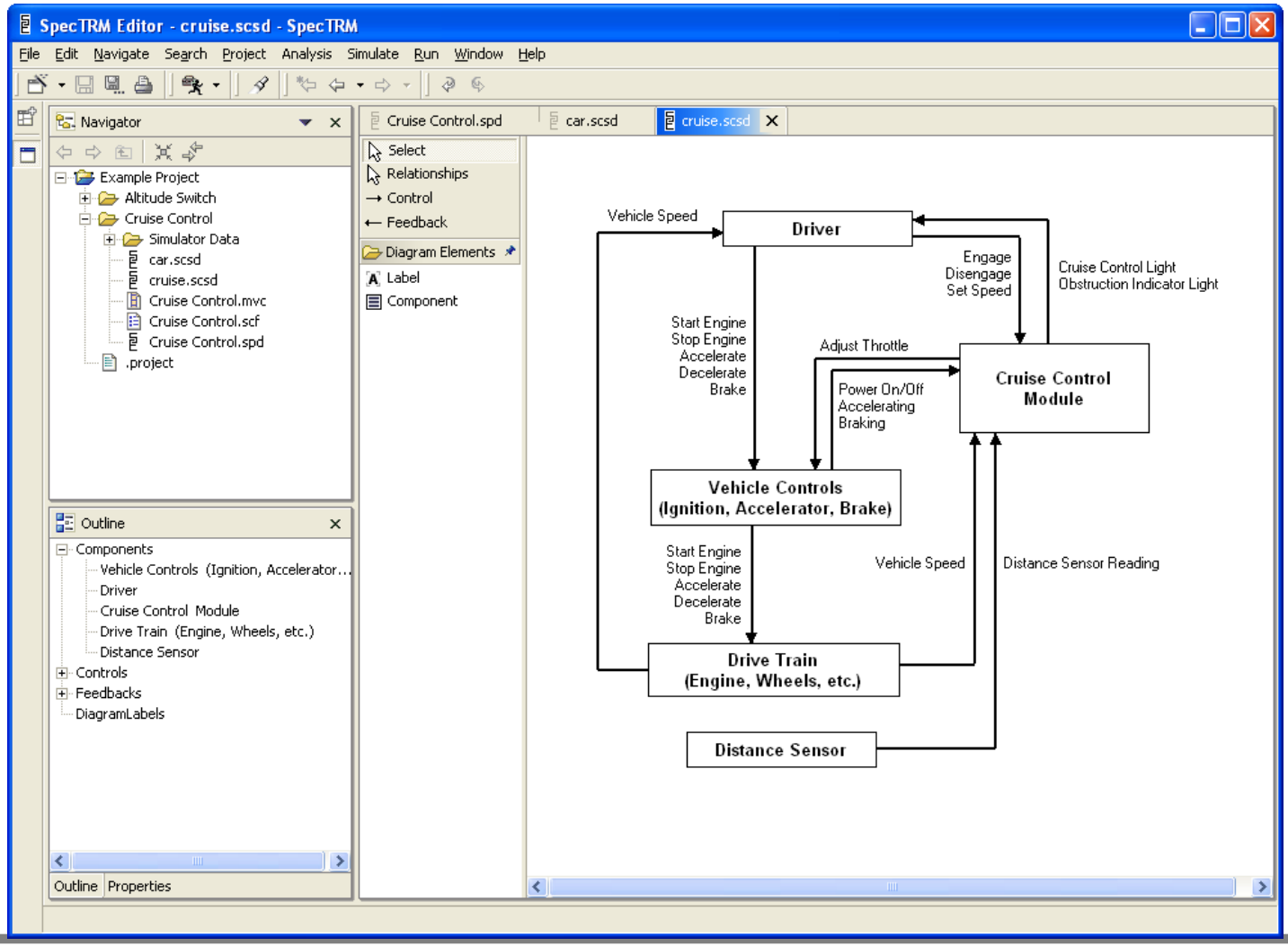


Formal mathematical representation:

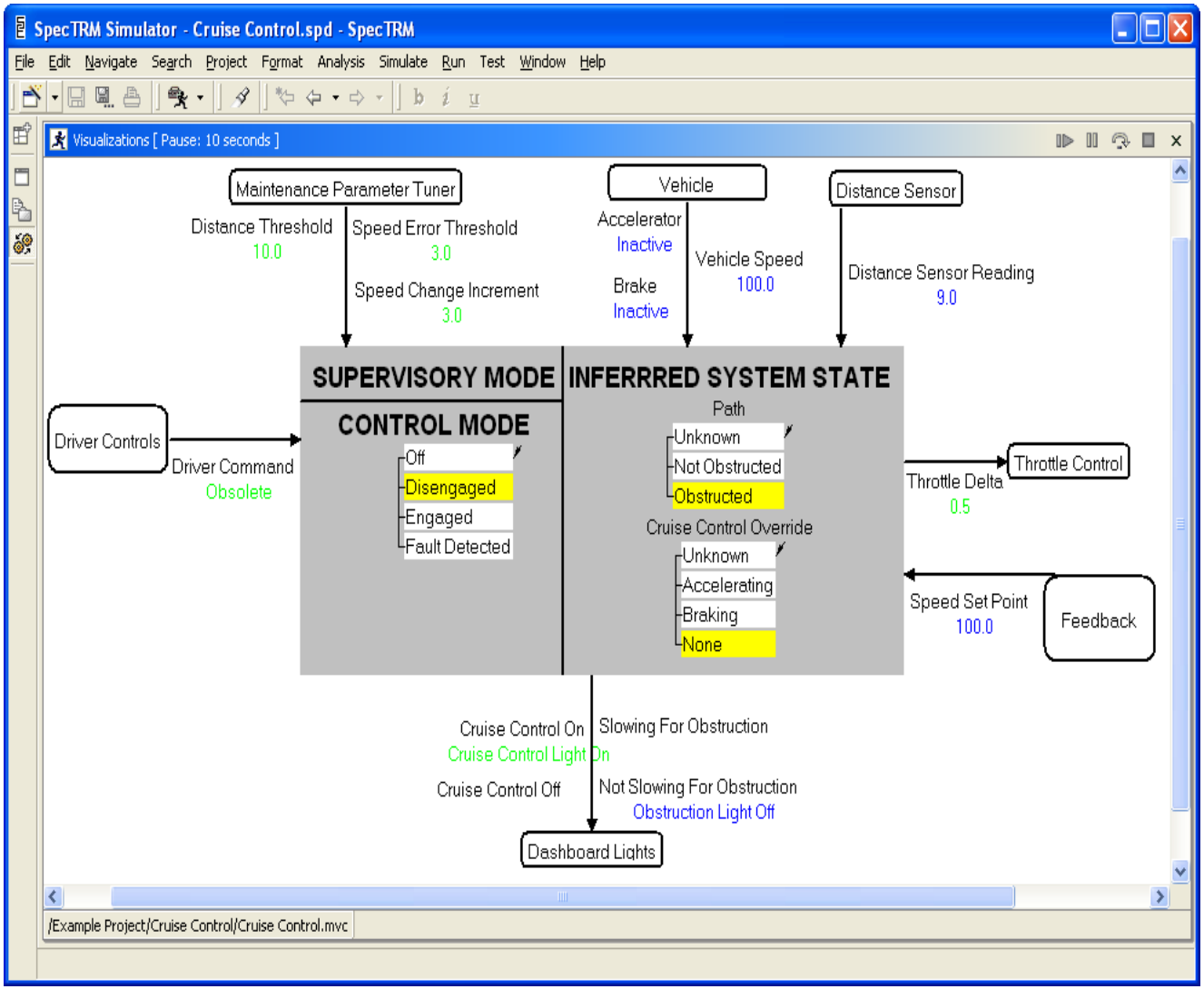
Other-Traffic =
(Alt-Reporting == Lost) ∧ ¬Bearing-Valid ∨ (Alt-Reporting == Lost) ∧ ¬Range-Valid ∨
(Alt-Reporting == Lost) ∧ Bearing-Valid ∧ Range-Valid ∧ ¬Proximate-Traffic-Condition ∧
¬Potential-Threat-Condition ∨ (Other-Aircraft == On-Ground)

(Leveson, 2000), (Zimmerman, 2002)

Hyperlinking to more detailed views of control structure



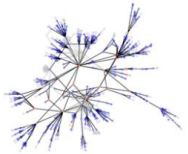
Requirements spec is executable!



Timeline view of simulation

Intent Specification

	Environment	Operator	System and components	V&V
Level 0 Prog. Mgmt.	Project management plans, status information, safety plan, etc.			
Level 1 System Purpose	Assumptions Constraints	Responsibilities Requirements I/F requirements	System goals, high–level requirements, design constraints, limitations	Preliminary Hazard Analysis, Reviews
Level 2 System Principles	External interfaces	Task analyses Task allocation Controls, displays	Logic principles, control laws, functional decomposition and allocation	Validation plan and results, System Hazard Analysis
Level 3 System Architecture	Environment models	Operator Task models HCI models	Blackbox functional models Interface specifications	Analysis plans and results, Subsystem Hazard Analysis
Level 4 Design Rep.		HCI design	Software and hardware design specs	Test plans and results
Level 5 Physical Rep.		GUI design, physical controls design	Software code, hardware assembly instructions	Test plans and results
Level 6 Operations	Audit procedures	Operator manuals Maintenance Training materials	Error reports, change requests, etc.	Performance monitoring and audits



Academic background

Hazard Causal Analysis

Hazard Causal Analysis Method

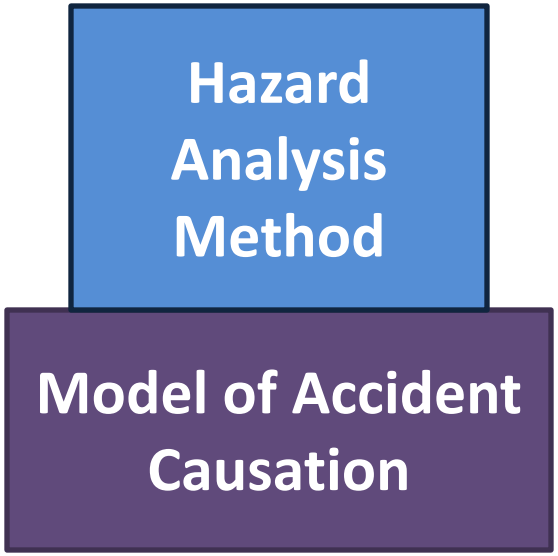
Model of Accident Causation

Accident Models

- Chain of events (1900s)
  - Accidents are caused by a sequence of events
  - Simple linear relationships between events
  - "break the chain"
- Parameter deviation (1960s)
  - Accidents are caused by parameter deviations
  - Ex: caused by no flow, too much pressure, etc.

(Heinrich, 1931); (Lawley , 1974); (Ladkin, 2005);

Traditional Hazard Analysis Methods



- Failure Modes Effects and Criticality Analysis (1949)
  - Reliability technique; start with failures, find effects
- Fault Tree Analysis (1961)
  - Top-down approach; start with hazard, find failure combinations
- Hazards and Operability Analysis (1960s)
  - Apply guidewords to components, find consequences
- Event Tree Analysis (1975)
  - Start with initiating event, trace forward in time

(Hammer, 1972); (Lawley , 1974); (Vesely et al, 1981); (Rasmussen, 1975); (Rasmussen, 1990)

Basis for a new foundation of safety engineering

Old Assumption	New Assumption
Safety is increased by increasing system or component **reliability**	High reliability is neither necessary nor sufficient for safety
Accidents are caused by **chains of directly related events**	Accidents are **complex processes** involving the **entire socio-technical system**
Probabilistic risk assessment based on event chains is the best way to assess and communicate safety and risk information	Not necessarily so!
Most accidents are caused by **operator error**.	Operator error is a **product of the enviroment** in which it occurs.
Highly reliable sotware is safe.	Not necessarily so!
Major accidents occur from the **chance** simultaneous occurrence of **random events**.	Systems will tend to **migrate towards states of higher risk**. This is predictable and preventable.
Assigning **blame** is necessary to learn from and prevent accidents or incidents.	Blame is the enemy of safety.

(Leveson, 2011)