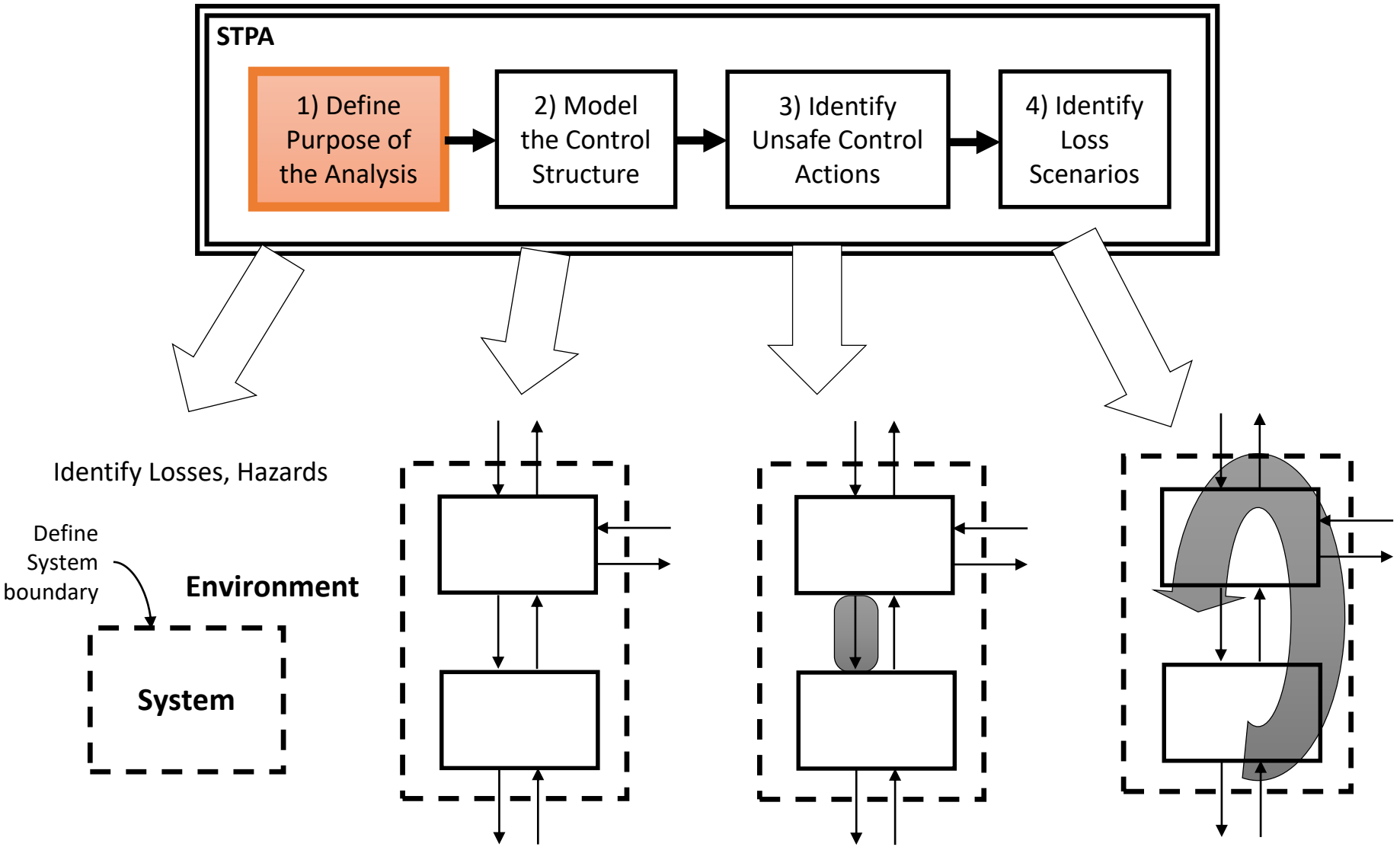
Massachusetts Institute of Technology

Use of STPA in Practice: Lessons Learned

Dr. John Thomas

Any questions? Please email: jthomas4@mit.edu

System-Theoretic Process Analysis (STPA)



Losses,
System-level Hazards

Incorrect Losses

- Loss of brake pressure
- Loss of engine RPM
- Loss of pressurizer pressure
- …

Incorrect System-level Hazards

System-level Hazards
Environmental hazard
Electrical hazard
Mechanical hazard
Slipping, tripping, and falling hazard
Hazards generated by materials and substances
Hazards generated by neglecting ergonomic principles in machine
Hazards generated by radiation
Thermal hazard
Hazards generated by vibration

See STPA Handbook

Confusing hazards with failures

Professionals who are experienced in other hazard analysis methods sometimes fall into the trap of writing STPA hazards describing potential deviations from specified technical functions or describing physical component failures. You may be familiar with traditional techniques that begin by searching for a set of deviations, faults, or functional failures in the technical system. To identify a broader set of causes in STPA, we cannot assume that the defined and specified functions are safe and correct, that human operators will perform as expected, that automated behaviors will not induce human error or confusion, that off-nominal cases will not occur, or that the technical design, specification, and requirements are correct. For example, the hazard "Controlled flight of aircraft into terrain" can be included in STPA while it may be omitted by efforts to examine only purely technical functional failures.

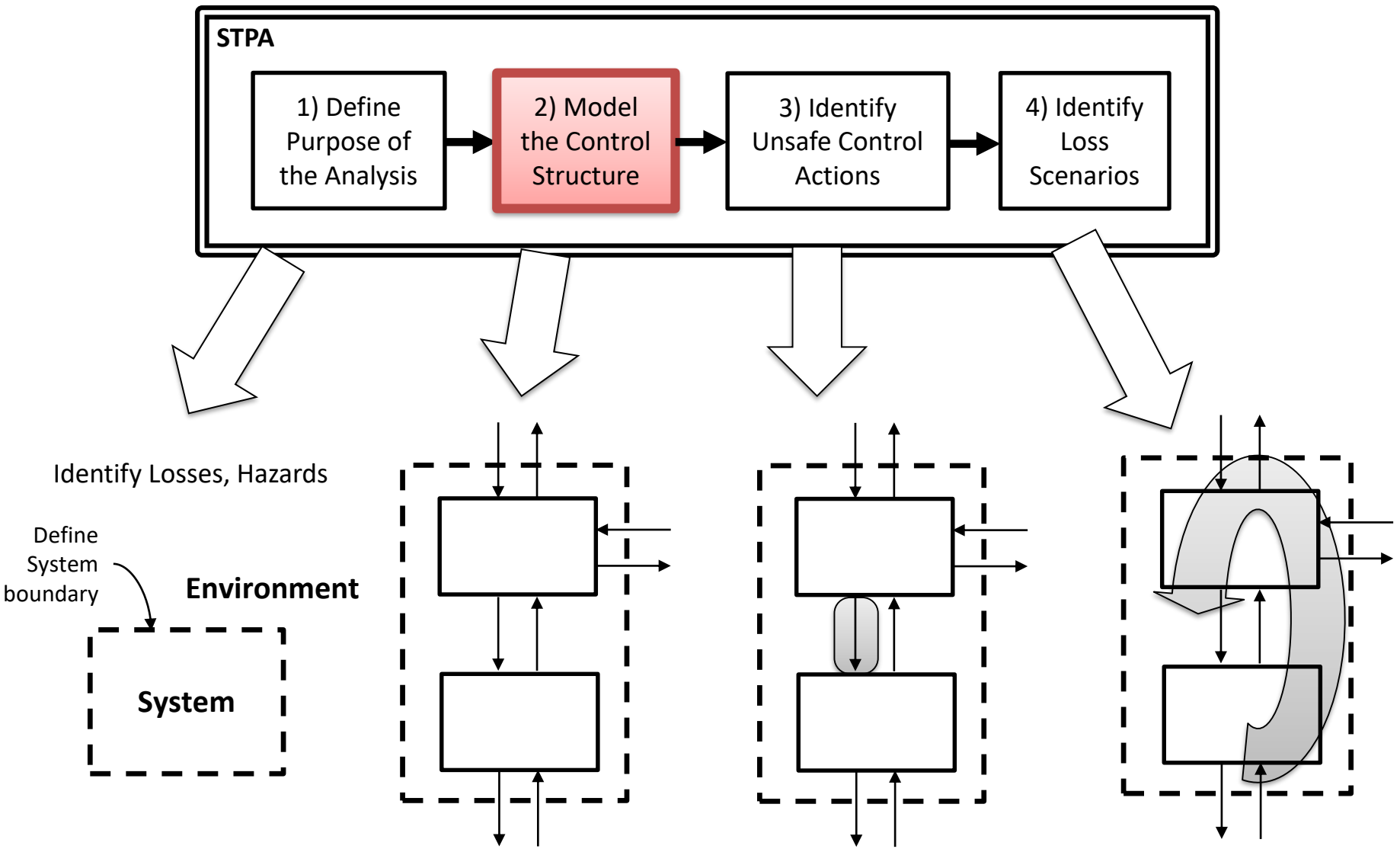
Hazard identification in STPA is about system states and conditions that are inherently unsafe—regardless of the cause. In fact, the system hazards should be specified at a high-enough level that does not distinguish between causes related to technical failures, design errors, flawed requirements, or human procedures and interactions.

What should I look for when reviewing hazards?

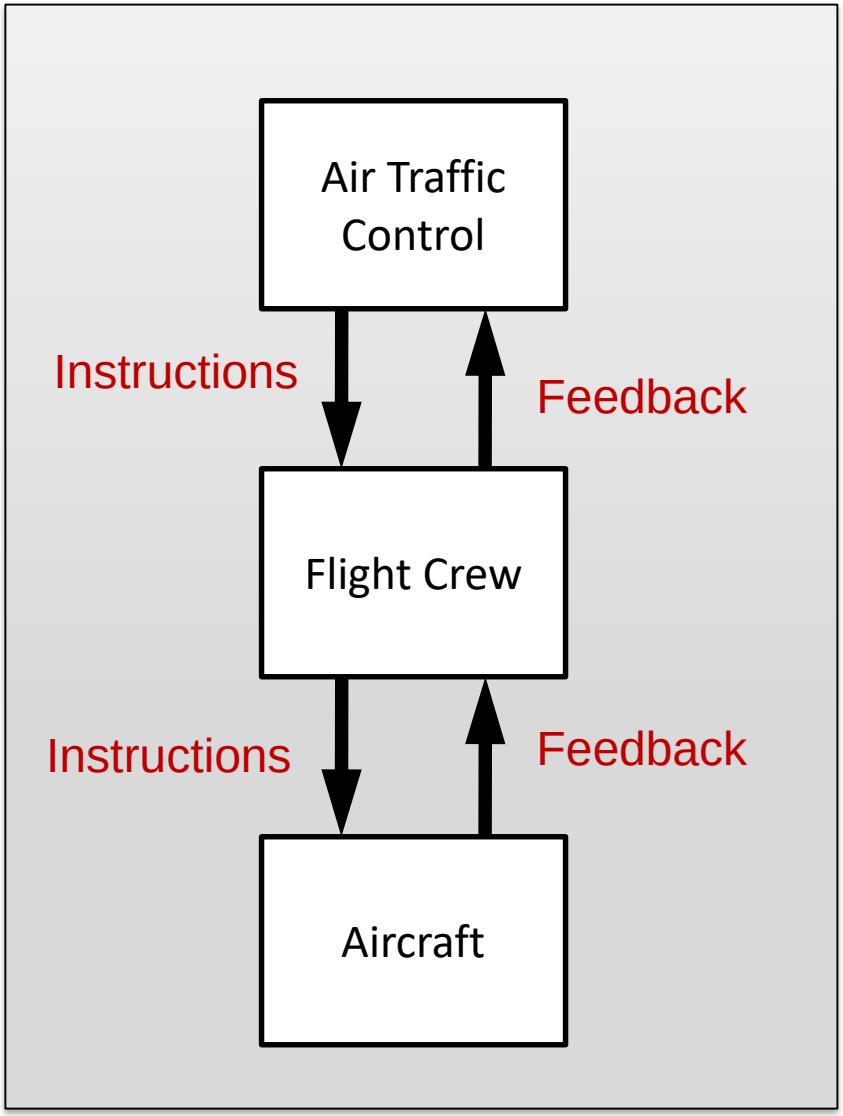
Tips to prevent common mistakes when identifying hazards

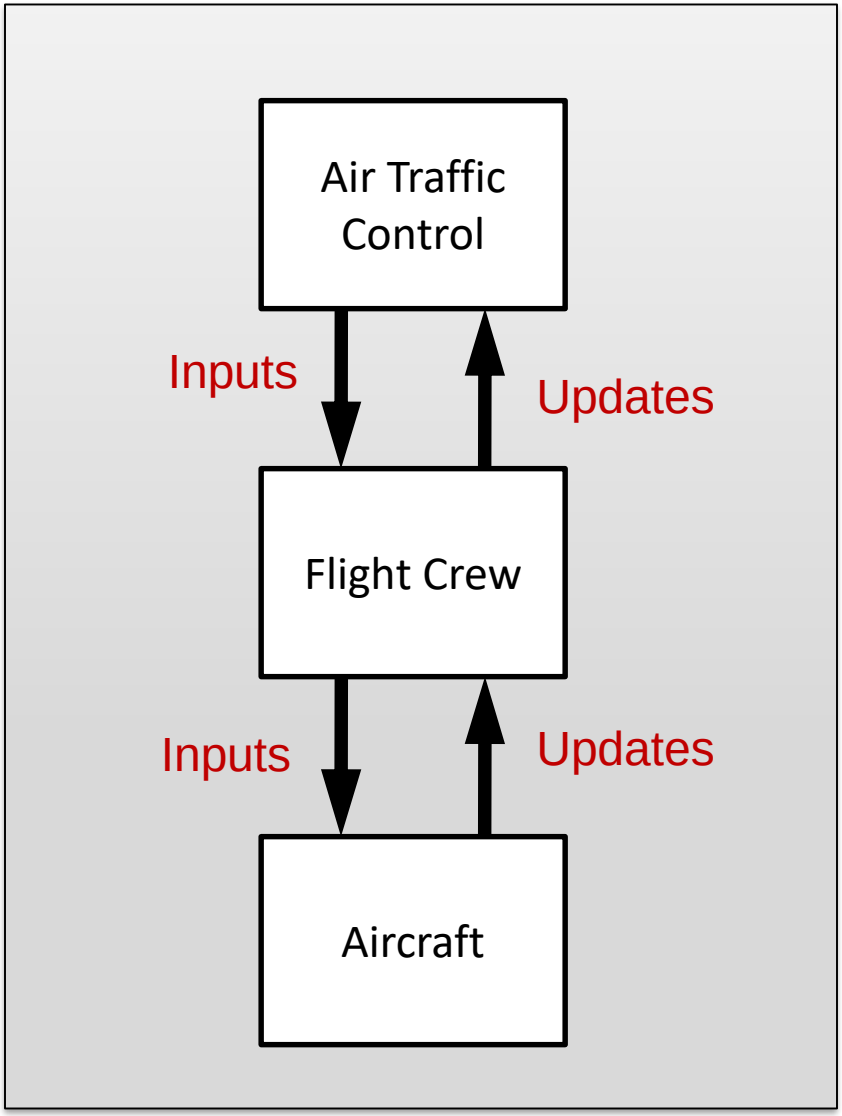
- Hazards should not refer to individual components of the system
- All hazards should refer to the overall system and system state
- Hazards should refer to factors that can be controlled or managed by the system designers and operators
- All hazards should describe system-level conditions to be prevented
- The number of hazards should be relatively small, usually no more than 7 to 10
- Hazards should not include ambiguous or recursive words like "unsafe", "unintended", "accidental", etc.

System-Theoretic Process Analysis (STPA)

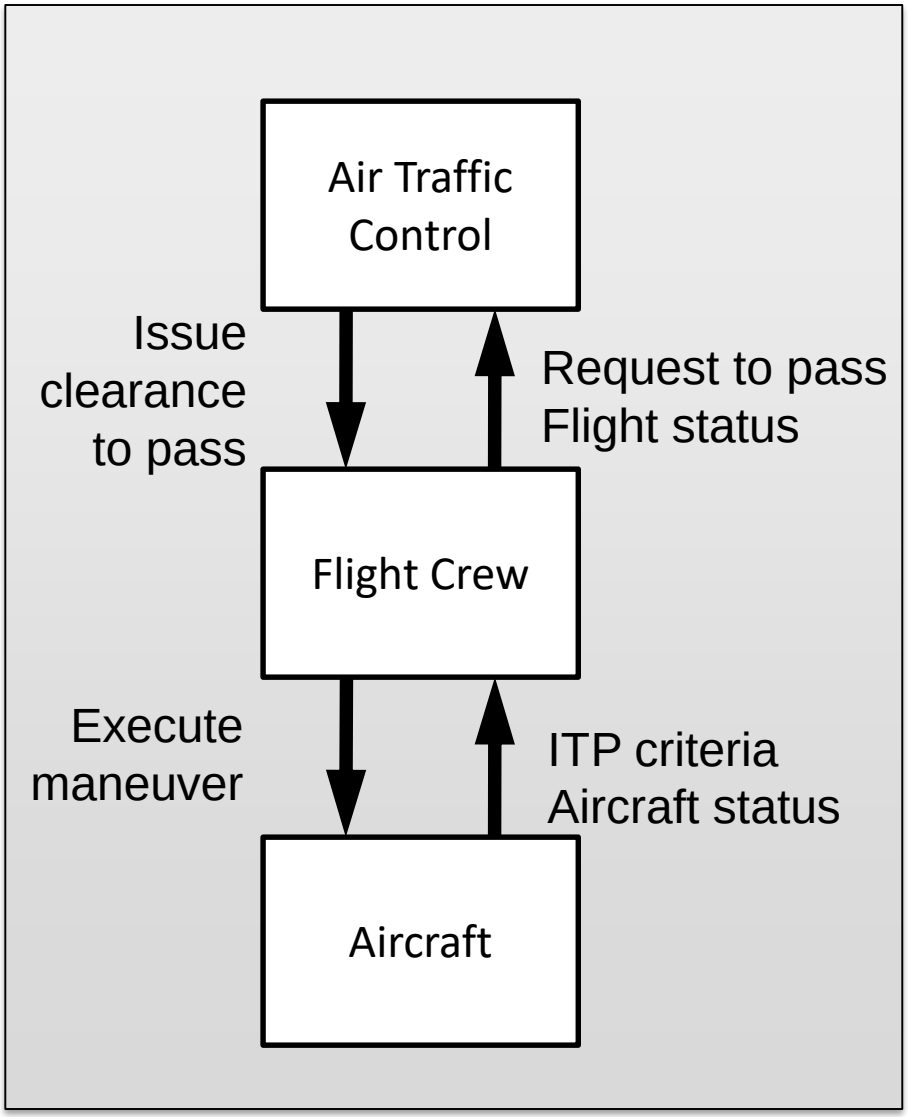


Control Structure that is too vague



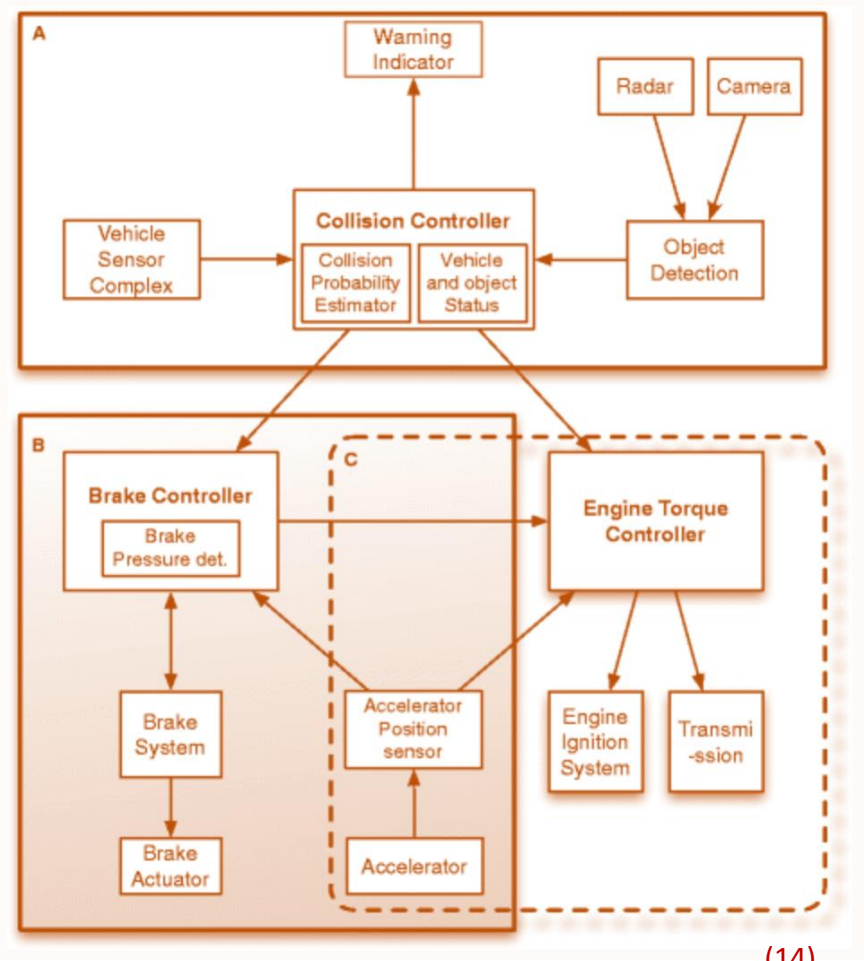


Better High-level Control Structure



- Note that “High-level” does not have to be vague!

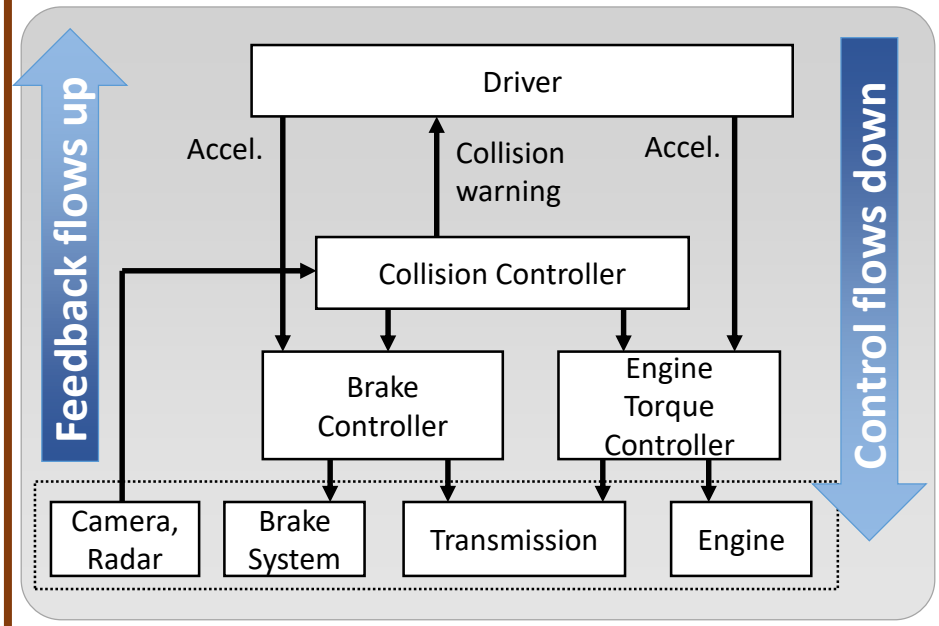
Incorrect control structure



(14)

- Missing or inconsistent control hierarchy
- Driver cmds, but no driver
- Sensors and actuators with no controller
- Controlled process?
- Control loops?

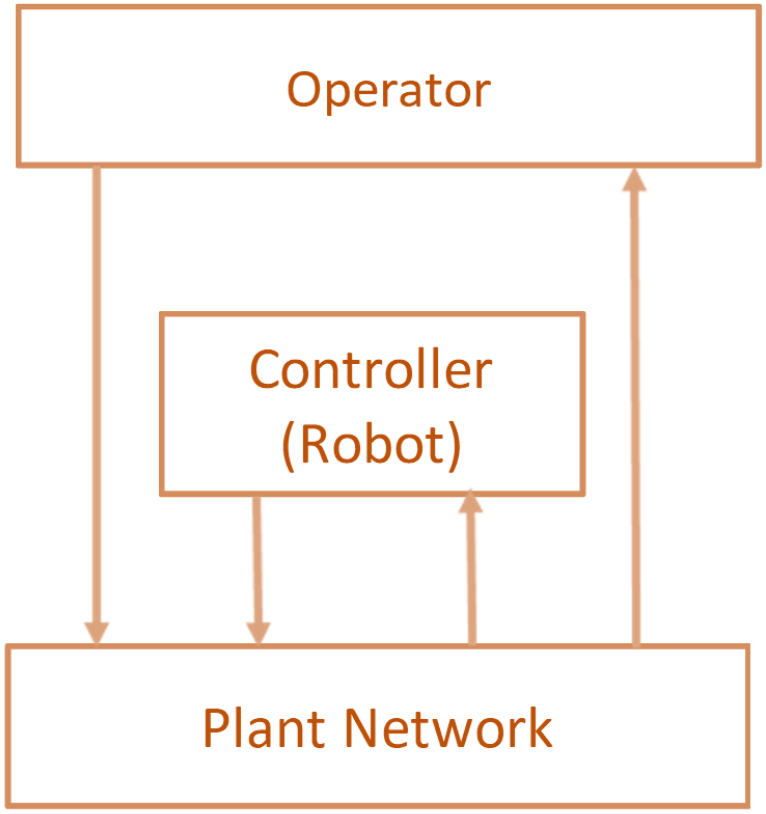
Better control structure (but incomplete)



(8)

- Defined control hierarchy
- Driver is included

Incorrect control structure



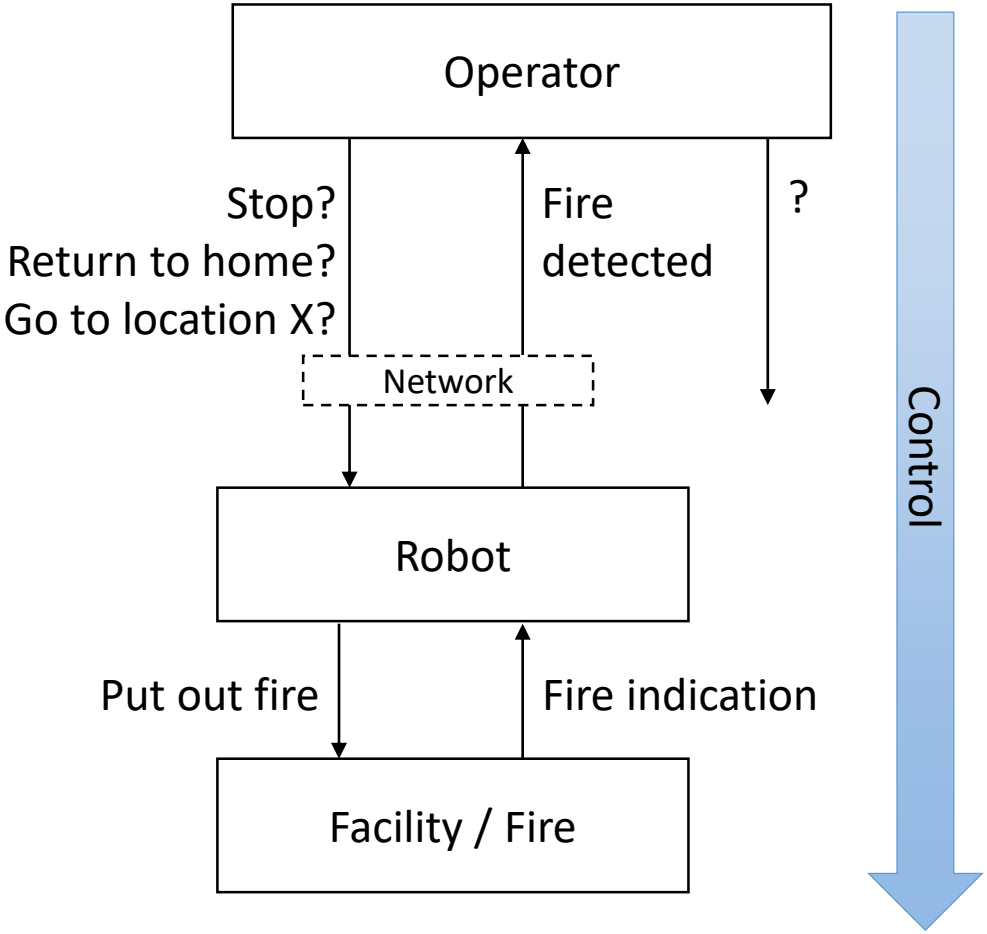
Control hierarchy?

Is the network really the ultimate controlled process

No commands to the Robot?

What are the commands/feedback?

Better control structure (but incomplete)



Properly defined control hierarchy

Controlled process is the facility/fire

Network is a “pass-through”, not generating its own control actions

See STPA Handbook

Common Points of Confusion

A control structure does not assume obedience

Do not confuse controllers and control actions with obedience. Just because a controller sends a control action does not mean that in practice it will always be followed. Likewise, just because a feedback path is included in a control structure does not mean that in practice the feedback will always be sent when needed or that it will be accurate. The control actions and feedback in a control structure simply indicate that a mechanism will be created to send this information (that is, it will be in the system design). It does not imply or assume anything about how controllers and processes will actually behave in practice. In fact, a major goal of STPA is to analyze the control structure and anticipate how each element might behave in unsafe and potentially unexpected ways.

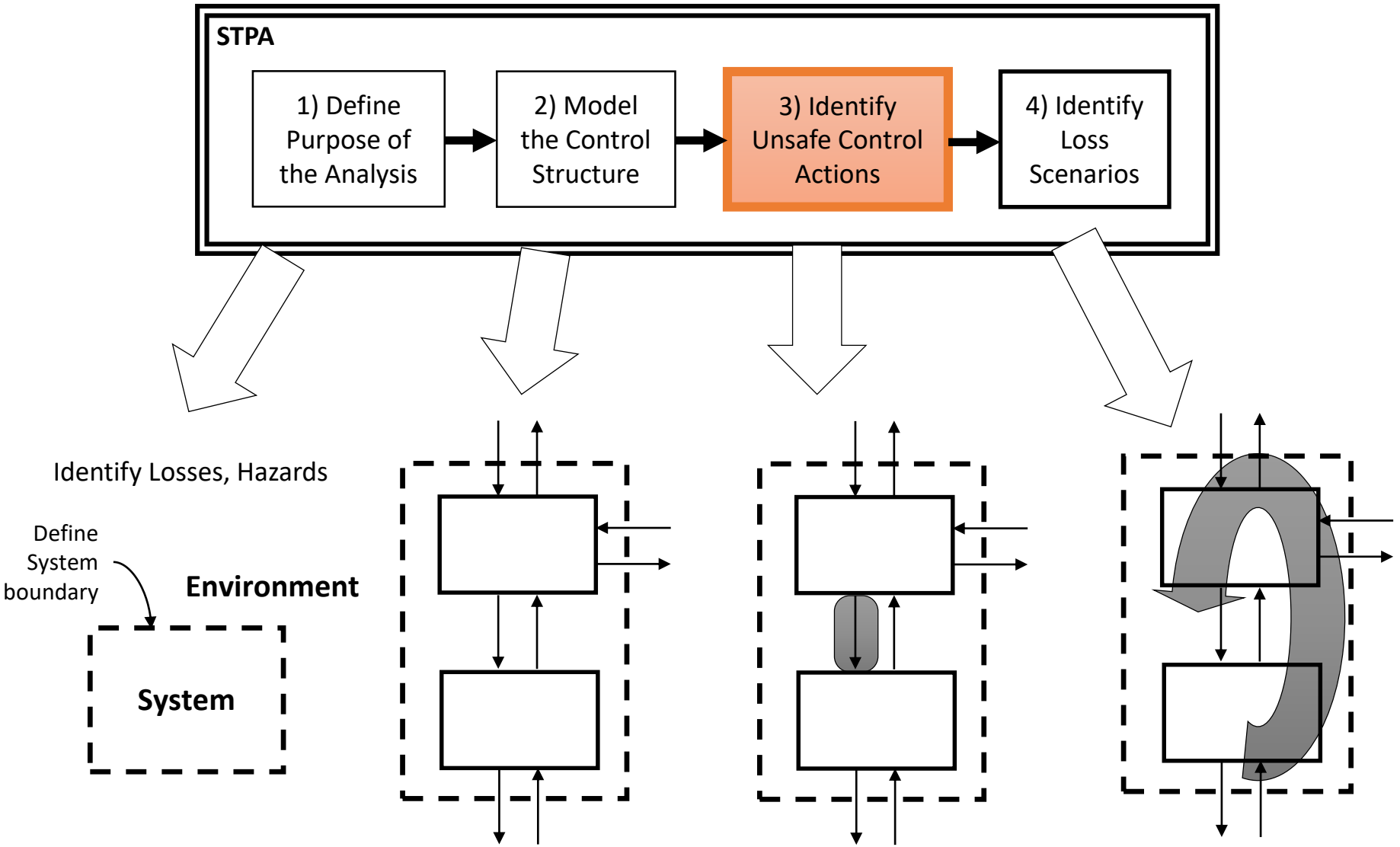
What should I look for when reviewing a control structure?

The following tips can help find common mistakes in a control structure:

Tips to prevent common mistakes in a control structure

- Ensure labels describe functional information that is sent, not a specific physical implementation.
- Avoid ambiguous and vague labels like simply "Command" or "Feedback" when the type of information is known.
- Check that every controlled physical process is controlled by one or more controllers (not always required, but often indicates a mistake).
- Review responsibilities (including traceability) for conflicts and gaps.
- Check that control actions needed to satisfy the responsibilities are included.
- Check that feedback needed to satisfy the responsibilities is included. (optional if applied early in concept development when feedback is unknown; later steps can identify missing feedback)

System-Theoretic Process Analysis (STPA)



Incorrect UCAs (Unsafe Control Actions)

- Pilot fails to recognize TCAS alert
- Does not monitor emergency brake operation
- Decreases funding

- Do not use “Fails” for human actions
- “Recognize” is not a control action
- “Monitor” is not a control action
- Missing the action
- Missing the context

Better UCA

UCA-1:

Pilot
does not provide
pitch up cmd
when conditions exist for climb RA
[H-1]

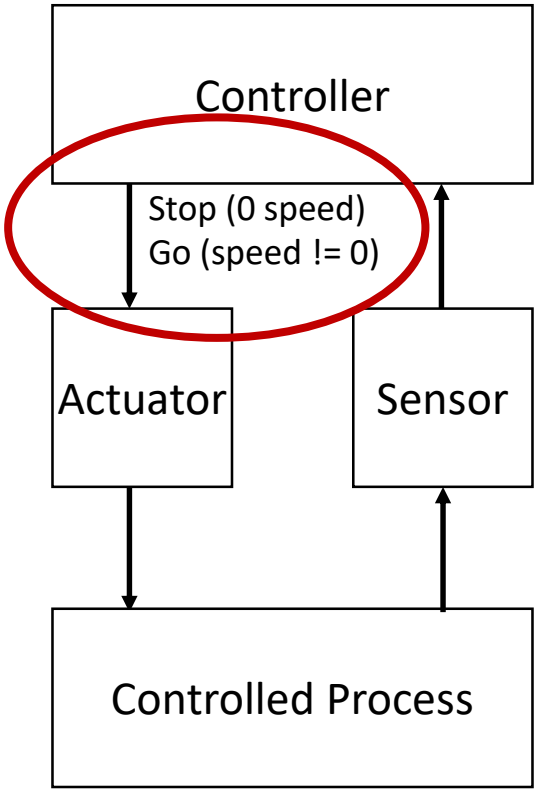
Includes all necessary UCA elements:

- Source controller
- Type
- Control Action
- Context
- Traceability to hazards

Tips for Specifying Unsafe Control Actions

- Start every UCA with the source controller
- A UCA is not just a statement about the state of a component
- A UCA is not just a statement about the outcome
- A UCA should include an observable output of the controller (an action or inaction)
  - Not a thought or a process like "monitoring“ or “recognizing”.
  - Look at arrows on the control structure
- Do not use the word “fail” in a UCA
  - These are not necessarily failures. They may or may not be caused by failures, and we may not know all the causes when STPA Step 3 is performed.

Incomplete UCAs



	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Applied too long, Stopped too soon
Go Cmd	Controller does not provide Go cmd when _____	Controller provides Go cmd when obstacle is in path	…	…
Stop Cmd	Controller does not provide Stop cmd when obstacle is in path	Controller provides Stop cmd when _____	…	…

Other UCAs are missing. Sub-categories:

- … Provides go with excessive speed…
- … Provides go with insufficient speed…
- … Provides go in opposite direction…
- … Provides go in unstable way (e.g. rapidly changing speed) …

Incorrect UCAs

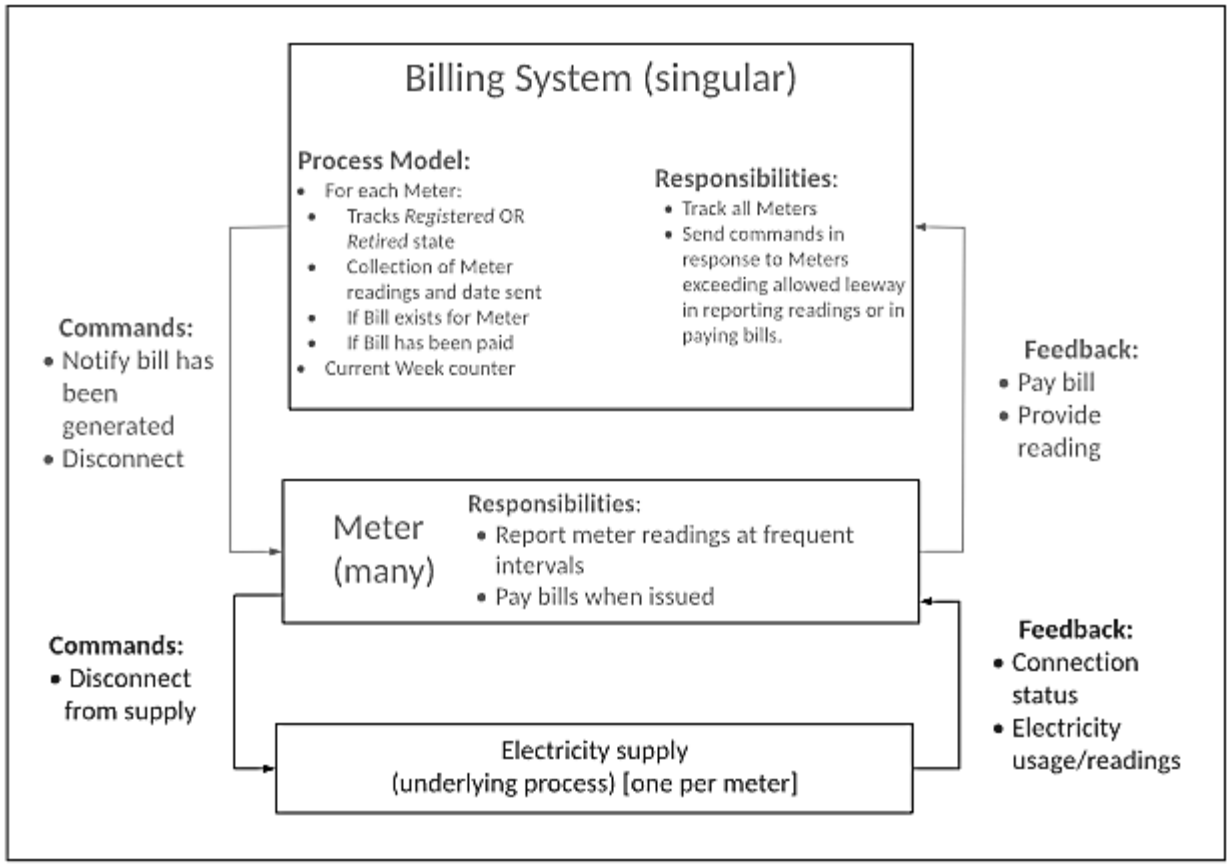


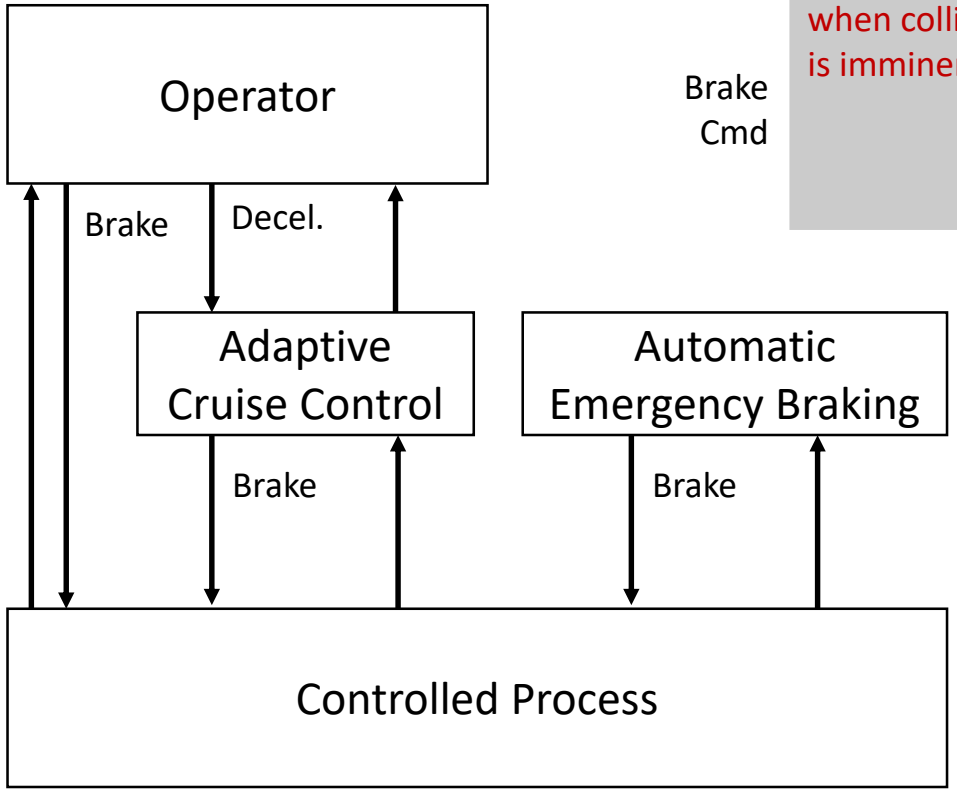
Figure 1: Functional control structure for smart meter example.

Control Action	*Is issued*	*Is not issued*	*Is issued out of sequence*	*Is issued for incorrect duration*
Register Meter	An invalid meter is re-registered.	A meter fails to be registered.	A meter is registered multiple times.	N/A - registration is discrete.

Table 2: Control action analysis results

Confusing control actions from multiple controllers

	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Applied too long, Stopped too soon
Brake Cmd	when collision is imminent	…	…	…



Author identified a valid UCA, but it was inadequately communicated to reviewers and others.

Confusion can be avoided by writing whole UCA.

UCA-1: Automatic Emergency Braking does not provide Brake Cmd when Collision is Imminent [H-1,2]

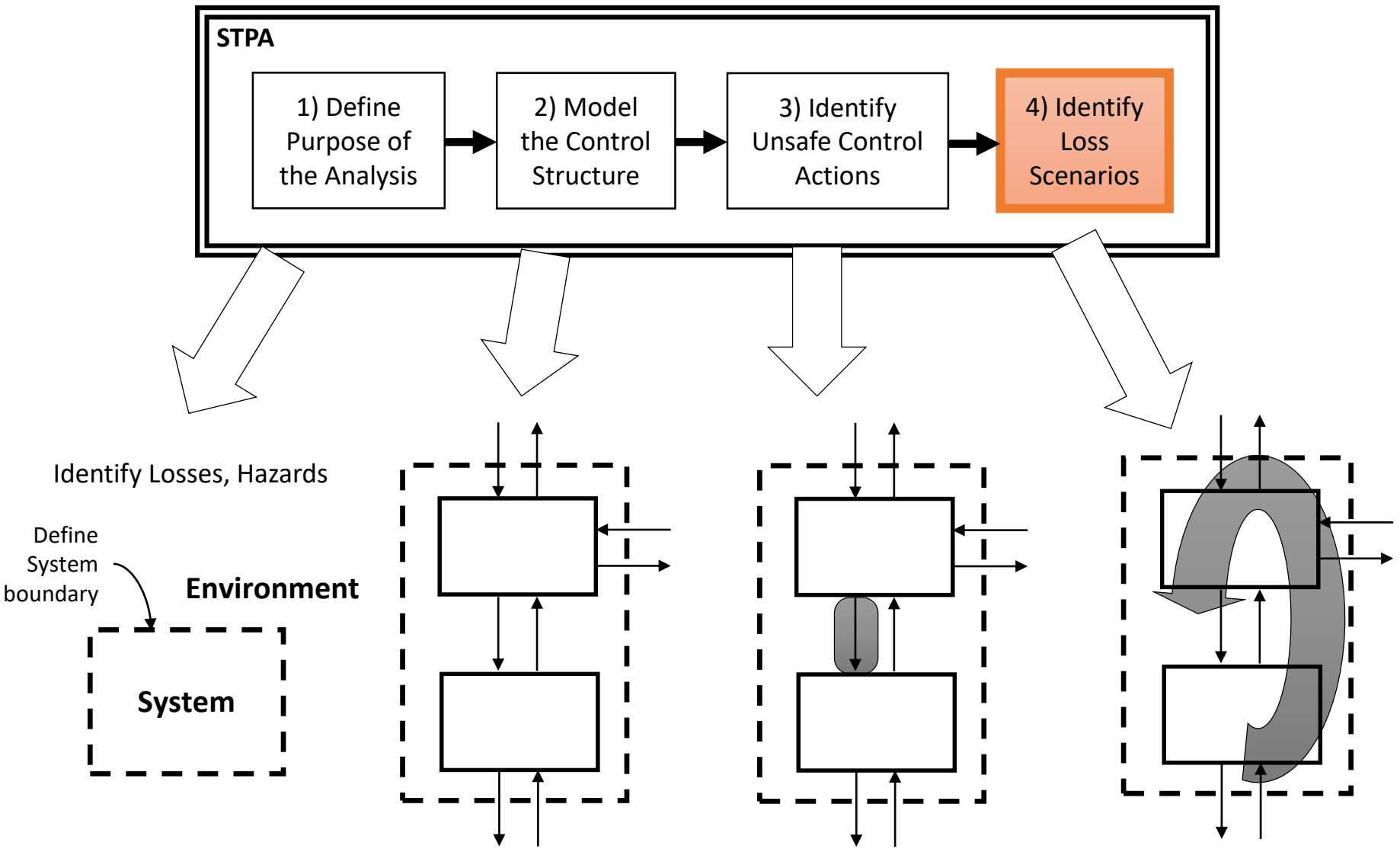
See STPA Handbook

Tips to prevent common mistakes when identifying UCAs

- Ensure every UCA specifies the context that makes the control action unsafe.
- Ensure UCA contexts specify the actual states or conditions that would make the control action unsafe, not potential beliefs about the actual states.
- Ensure the UCA contexts are defined clearly.
- Ensure the UCA contexts are included and not replaced by future effects or outcomes.
- Ensure traceability is documented to link every UCA with one or more hazards.
- Review any control action types assumed to be N/A, and verify they are not applicable.
- For any continuous control actions with a parameter, ensure that excessive, insufficient, and wrong direction of the parameters are considered.
- Ensure any assumptions or special reasoning behind the UCAs are documented

STPA Handbook

System-Theoretic Process Analysis (STPA)



Incorrect causal factors

Step 1 no.	Hazards	Severity	Causal factors
1a	System dysfunction due to failure of object detection system	Catastrophic	Object detection component failure (camera, radar, or motion sensors)
			Communication error (no signal)
1b	Malfunctioning of the system due to incorrect input from object detection system	Catastrophic	Corrupted communication (wrong signal)
			Malfunctioning of camera, radar, and motion sensors
			Communication system does not work on time
2a	Incorrect and missing calculation of vehicle status and collision probability due to failure or malfunctioning of vehicle complex sensors	Catastrophic	Failure of vehicle sensors

- These are individual factors only, no combinations
- The connection to UCAs and Hazards is not clear
- These only consider failures and malfunctions, not components working as required, as designed

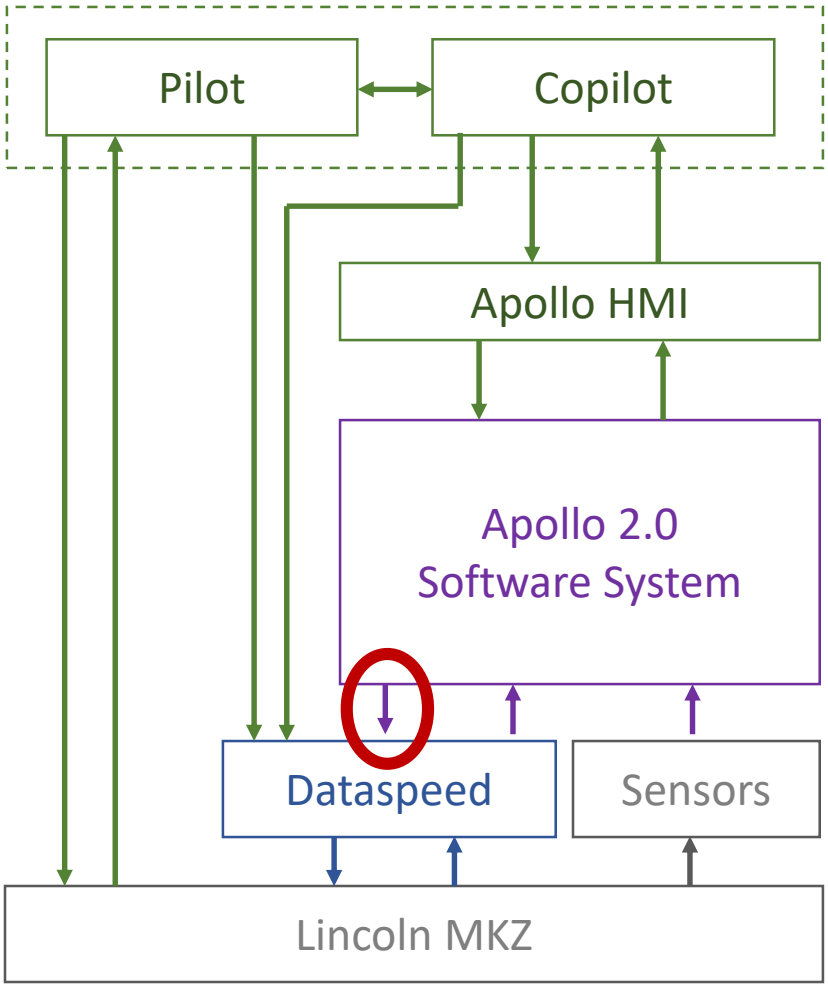
See STPA Handbook

Tips to prevent common mistakes when identifying Scenarios

The most common mistake is to identify individual causal factors rather than a scenario. For example, you may be tempted to create list of factors like “wheel speed sensor failure”, “wheel speed feedback is delayed”, “loss of power”, etc. The problem with listing individual factors outside the context of a scenario is that it’s easy to overlook how several factors interact with each other, you can overlook non-trivial and non-obvious factors that indirectly lead to UCAs and hazards, and you may not consider how combinations of factors can lead to a hazard. Considering single factors essentially reduces to a FMEA where only single component failures are considered.

STPA Handbook

Better STPA Scenario Example



UCA-1: Apollo provides throttle cmd when forward collision is imminent

- Can occur if Apollo incorrectly believes forward collision is not imminent (Process Model Flaw)
- Feedback: Apollo is not designed to detect automatic emergency braking or disable throttle commands.

Resulting potential requirements

- R-1: Apollo must not provide throttle cmd when AEB engages
- ..

Actual design: The vehicle is designed to override automatic emergency braking if throttle commands are received

Better STPA Scenario Example

Flawed Process Model:
ISS Crew incorrectly believes HTV is not approaching ISS

UCA-2: ISS Crew provides Free Drift Cmd when HTV approaching ISS