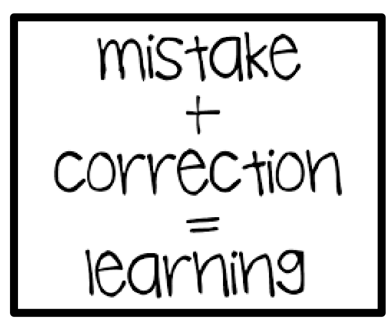
Common ~~Mistaeks~~ Using STAMP and Its Tools

Nancy Leveson

MIT

Performing STPA



Defining System Hazards

- Most common mistake is in process of defining system hazards
  – Should only be at system level
  – Usually only a few (less than 10-15)
- Must be within the scope of the system (under system designers' control)

Defining Hazards (2)

- System-level hazards <u>do NOT mention components</u>

  Examples:

  <u>Correct</u>: Loss of control of aircraft

  <u>Incorrect</u>: Pilot does not maintain control of aircraft

  Elevators do not control pitch

  Control surface failure

- Narrows inquiry too much, start by looking at components and miss big picture (the entire system)
- To take a system view, must start at system level
  - Will trace system hazards to components later in analysis

Correct Losses and Hazards for An Example

- Loss and system hazard

  Loss: Spacecraft lost

  Hazard: Spacecraft has inadequate heat and power

  Spacecraft destroyed while landing on planet

  Spacecraft hit by space debris

Incorrect Examples

- Incorrect System Hazards

  For nuclear powered spacecraft:

  - Turbine generates less mechanical energy than needed.
    - Only one part of the nuclear power system
    - Will later draw control structure and this may be one of the UCAs

  The ones below are all causes of the first (causal scenarios)

  - Turbine does not rotate.
  - The steam generator provides low steam flow in the turbine inlet line
  - Broken disk
  - Steam impaired lubrication of bearings causing wear on bearings
  - The steam generator sends liquid along with the air to the turbine
  - Too much wear on the bearings
  - Human error

Defining Hazards (3)

- Use of the word “failure” (turns problem into reliability)
  - “System failure” provides no useful information or goal for analysis
    - Always multiple requirements and constraints
    - Often tradeoffs and conflicts
  - Examples of incorrect hazards

    “Chemical plant fails”

    (instead) Plant releases toxic chemicals into the environment around the plant

    “Software failed”

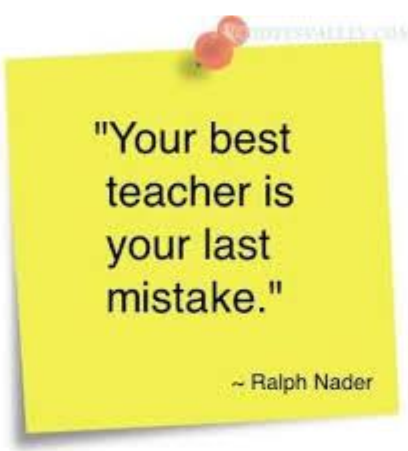
    “Pilots failed to maintain control of aircraft”

Stopping After Building a Control Structure

- Stop after create model and don’t do analysis
- Often useful but not STPA or CAST
- For STPA, need to identify all paths to the hazard (causal scenarios)
- For CAST, need to identify inadequate controls (flawed mental models and contextual factors in the loss)

Identifying Unsafe Control Actions

- **Incorrect**: put failures (or hazards) in table
  - Doing a FMEA (FMECA) or FTA using the STPA format
- **Correct**: table entries are the context in which control action leads to a hazard
- Remember, this is a paradigm change: it will require effort on your part at first to change the way you now think and do HA



Incorrect UCA Table

Control Action BSCU:	Not providing causes hazard	Providing causes hazard	Too soon, too late, out of sequence	Stopped too soon, applied too long
BSCU.1 Brake command	Brake command not provided [H4-1, H4-5]	Braking commanded excessively [H4-1, H4-5]	Braking commanded too late [H4-1, H4-5]	Brake command stops too soon [H4-1, H4-5]
	Brake command not provided [H4-1, H4-5]	Braking command provided inappropriately, [H4-1, H4-2, H4-5]	BSCU.1c2 Brake command applied more than TBD seconds, [H4-1, H4-5]	BSCU.1d2 Brake command applied too long (more than TBD seconds) [H4-1]

5 Parts of an Unsafe Control Action (Hazard)

1. Source Controller (Pilot, PVI, Automatic Controllers)
2. Control Action
3. Type of Unsafe Control (provided, not provided, wrong timing/order, wrong duration)
4. Context in which control action is unsafe
5. Consequences (system-level hazardous behavior)

BSCU: Braking command not provided during landing roll, resulting in insufficient deceleration and potential overshoot

5 Parts of an Unsafe Control Action (Hazard)

1. Source Controller
2. Control Action
3. Type of Unsafe Control
4. Context in which control action is unsafe
5. Consequences (system-level hazardous behavior)

Control Action BSCU:	Not providing causes hazard	Providing causes hazard	Too soon, too late, out of sequence	Stopped too soon, applied too long
BSCU.1 **Brake command**	BSCU.1a1 Brake command not provided **during RTO (to V1)**, resulting in inability to stop within available runway length **[H4-1, H4-5]**	BSCU.1b1 Braking commanded excessively **during landing roll**, resulting in rapid deceleration, loss of control, occupant injury **[H4-1, H4-5]**	BSCU.1c1 Braking commanded **before touchdown**, resulting in tire burst, loss of control, injury, **H4**other damage **[-1, H4-5]**	BSCU.1d1 Brake command stops **during landing roll before TBD taxi speed attained**, causing reduced deceleration **[H4-1, H4-5]**

Correct UCA (Context) Table

Control Action BSCU:	Not providing causes hazard	Providing causes hazard	Too soon, too late, out of sequence	Stopped too soon, applied too long
BSCU.1 Brake command	Brake command not provided **during RTO (to V1)**, resulting in inability to stop within available runway length **[H4-1, H4-5]**	Braking commanded excessively **during landing roll**, resulting in rapid deceleration, loss of control, occupant injury **[H4-1, H4-5]**	Braking commanded **before touchdown**, resulting in tire burst, loss of control, injury, other damage **[H4-1, H4-5]**	Brake command stops **during landing roll before TBD taxi speed attained**, causing reduced deceleration **[H4-1, H4-5]**

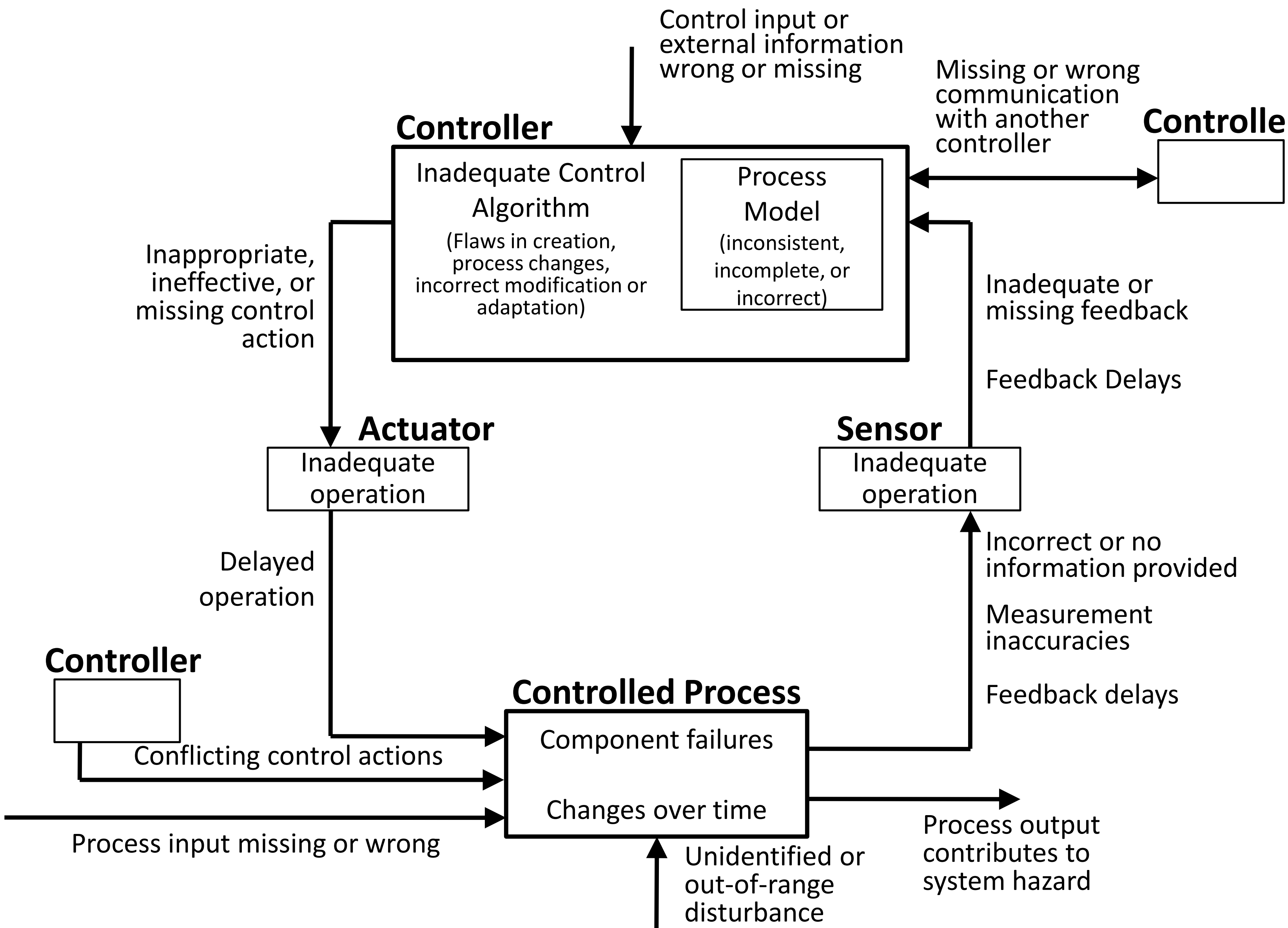
Correct UCA (Context) Table

Control Action BSCU:	Not providing causes hazard	Providing causes hazard	Too soon, too late, out of sequence	Stopped too soon, applied too long
BSCU.1 Brake command	**during RTO (to V1)**, **[H4-1, H4-5]**	**during landing roll**, **[H4-1, H4-5]**	**before touchdown, [H4-1, H4-5]**	**during landing roll before TBD taxi speed attained [H4-1, H4-5]**
	BSCU.1a2 **during landing roll**, **[H4-1, H4-5]**	BSCU.1b2 **during takeoff, [H4-1, H4-2, H4-5]**	BSCU.1c2 **after touchdown, [H4-1, H4-5]**	BSCU.1d2 **during landing roll [H4-1]**

Generating Causal Scenarios

- Not going to be accomplished in a few days
  - Takes months and even years to generate scenarios using traditional hazard analysis
  - STPA is more efficient and much less costly (maybe weeks), but it will never be trivial (even with new approaches)
  - If it were trivial, then engineering design would be trivial and we can replace all engineers with computers
- Sometimes oversimplify
  - CAST: Need to understand "why" someone thought it was the right thing to do
  - STPA: Need to identify all scenarios and get enough information to eliminate or mitigate the hazard (unsafe control)

Why might software open catalyst valve when water valve not open? *[Hint: Start with Process Model]*



Some Reasons for Incorrect Process Model

- Previously issued an Open Water Valve command but valve did not open (jammed, failed, etc.)
- Assumed that command had been executed. Why?

  i. No feedback about effect of previous command
  (<u>Control</u>: put feedback in design)

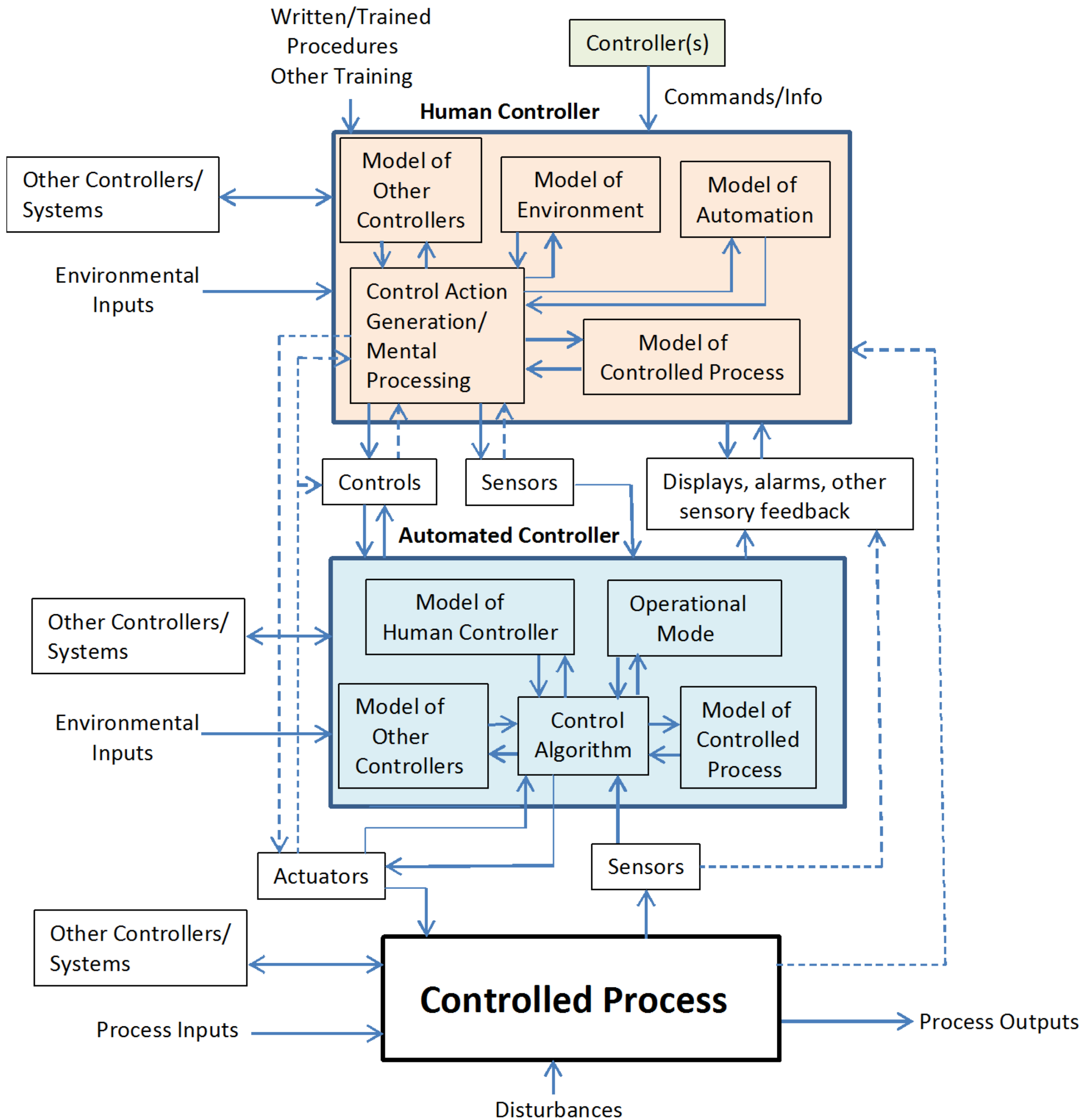
  ii. Feedback not received. [could go on to determine “why” here if want]
  (<u>Control</u>: Assume not executed)

  iii. Feedback delayed (could go on to determine “why” if want)
  (<u>Control</u>: wait predetermined time and then assume not opened)

  iv. Incorrect feedback received. Why? (maybe assumed that if reached valve, it would open [design error]
  (<u>Control</u>: add flow meter to detect water flow through pipe)

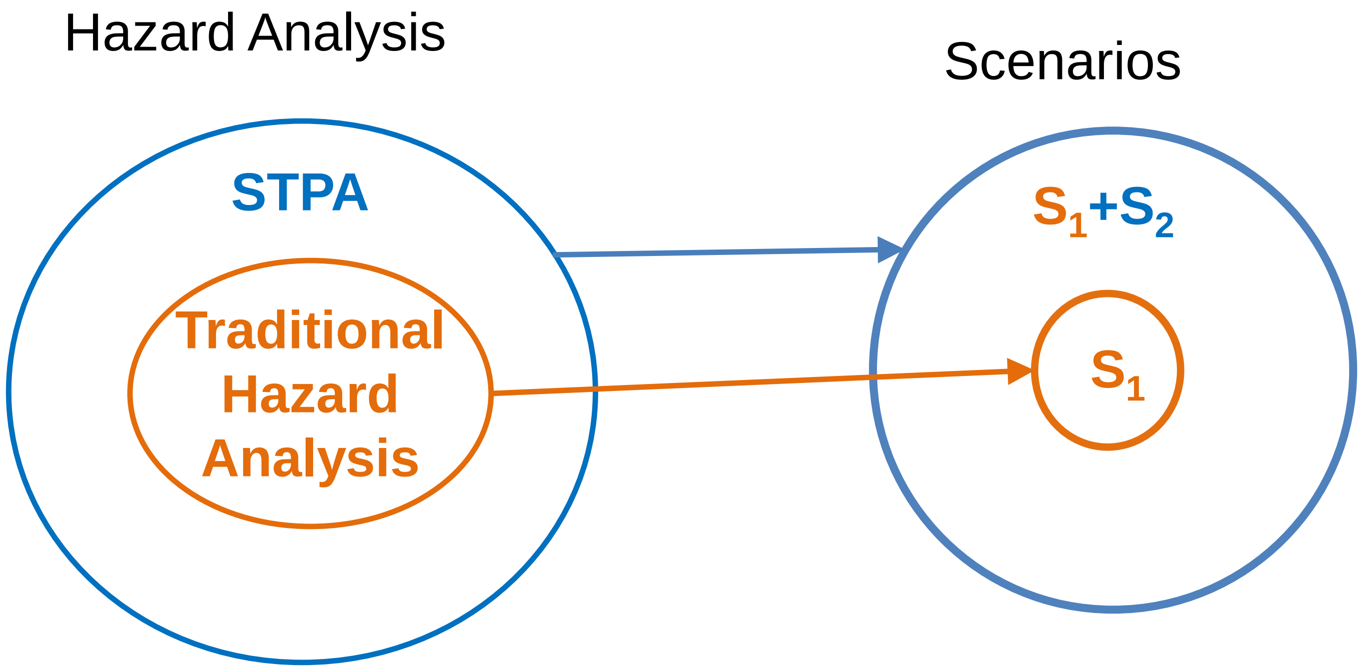
  etc.

Improved Model



Combining STPA with Old Analysis Techniques

- A tremendous waste of resources



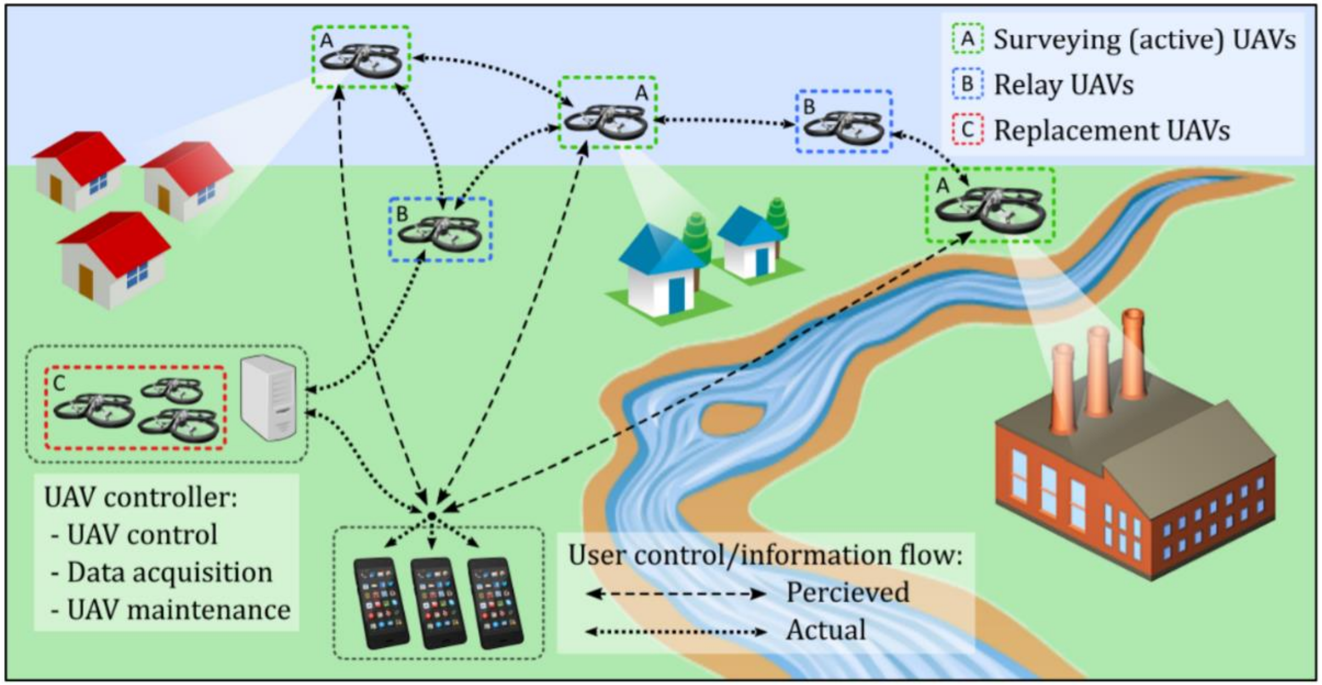
Trying to Apply STPA to Other Types of Models

- STPA is an analysis technique performed on a functional control structure

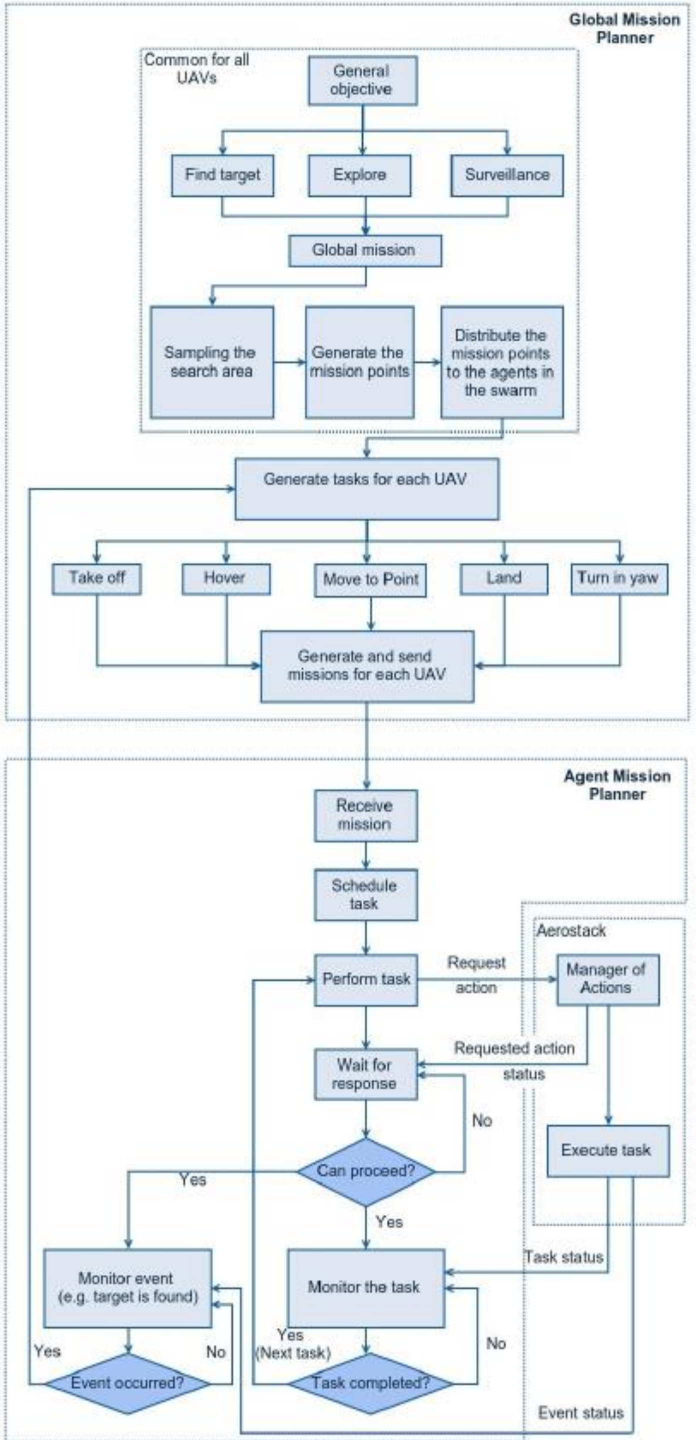
- Older HA techniques either do not have a model
  - Done on model in analyst's head, e.g., FTA
  - Or done on a different type of model (HAZOP)

- We need to add new functional control models to MBSE
  - Cannot just do STPA on UML, SysML, or other architectural models (and limit to one architectural style, i.e., OOD)
  - The model and analysis go together

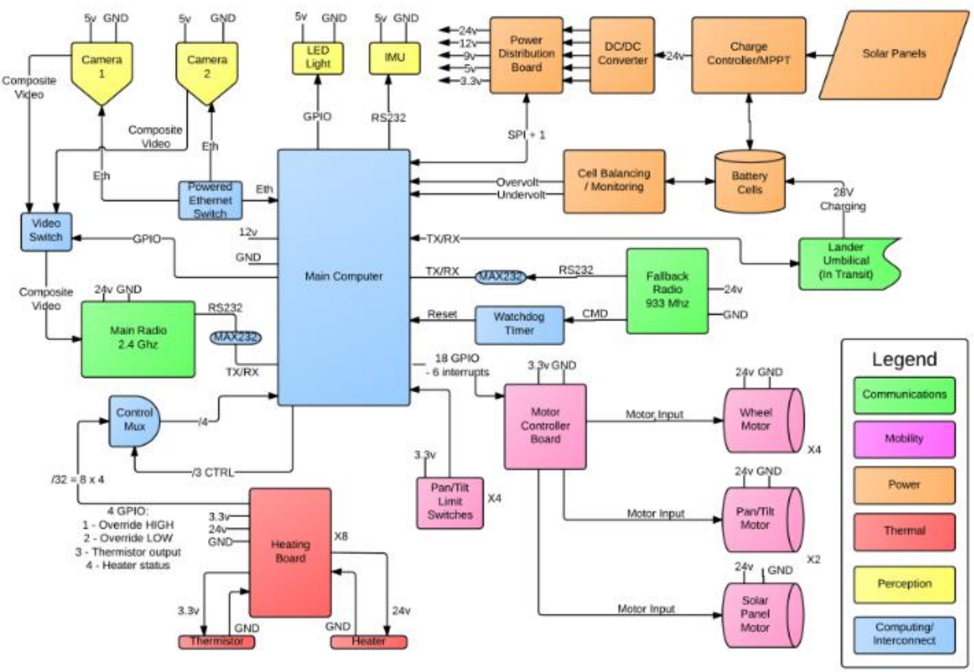
Information Flow Model



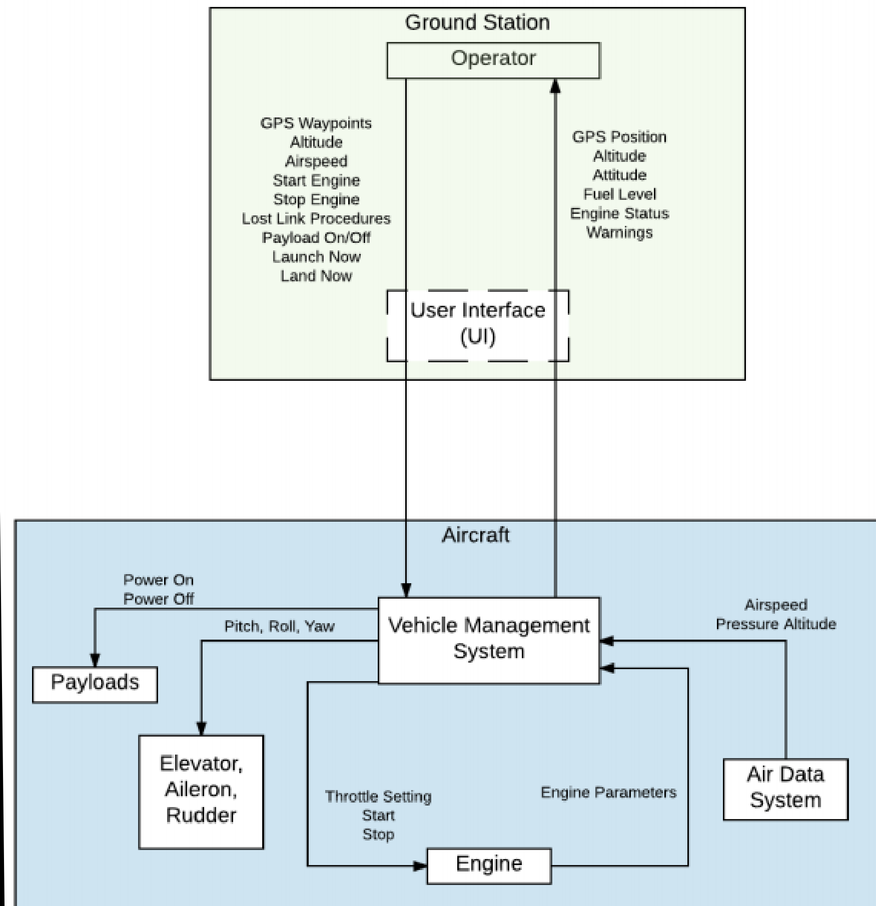
Mission Planning Communication Architecture (task/control flow model)



UAV Physical Data Structure



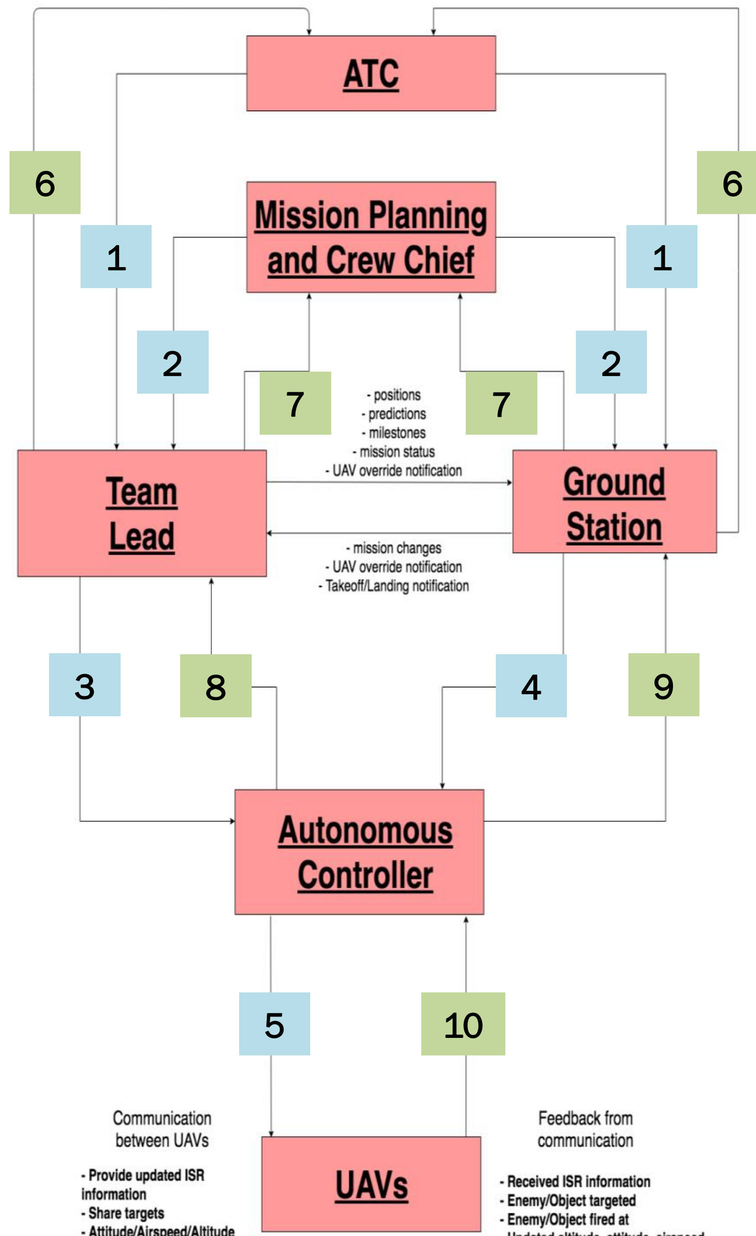
STPA: UAV Functional Control Structure



AFRL MUM-T Control Structure (UxAS)

Control Actions

1	Grant clearance Issue ground operation Issue approach/departure instructions
2	Issue mission plan Issue updates/changes
3	Surveil a region Search for a target Identify/Assess target Track target Aim/Fix on target Fire at/Engage target Send formation command Send override command
4	Takeoff Land Send override command Send altitude command Send airspeed command Start engine Stop engine Turn on payload Turn off payload Apply lost link procedure
5	Change altitude Set throttle



Feedback

6	Request takeoff/landing clearance Request ground clearance Confirm guidance/instructions
7	Mission Status
8	Visual Communication Checks Position Data Mission Progress Updates Waypoint destination Time to destination
9	Engine status Communication checks Current services status + warnings Mission Progress Updates Mission objectives Position Data Aircraft hardware status Waypoint destination Time to destination Unauthorized requests Lost link successful
10	GPS position Altitude Airspeed Fuel level Engine status Warnings/Cautions

Summary

- STAMP and tools represent a paradigm change
  - Will take some getting used to it
  - Hardest for people who have intensively used the traditional techniques
  - Easiest for system engineers and students because no "unlearning" required
- But benefits of using them are very high
  - More causal scenarios and additional types
  - Costs much less to do once learn it
  - Ability to use earlier in life cycle results in large ROI
- Will need ways to educate and integrate into organizations (e.g., facilitators, "teach the teachers")