STPA
Systems Theoretic Process Analysis

Agenda

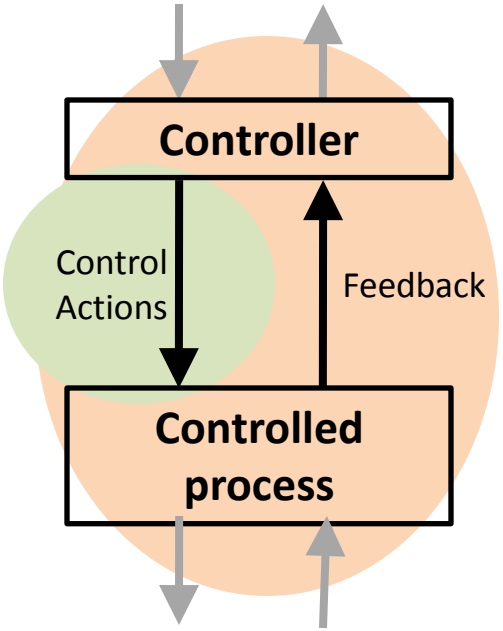
- Rigorous method for STPA Step 1
- STPA Step 2

STPA
(System-Theoretic Process Analysis)

STPA Hazard Analysis

STAMP Model

- Identify the hazards
- Define the control structure
- Step 1: Identify unsafe control actions, safety constraints
- Step 2: Identify causal factors, accident scenarios



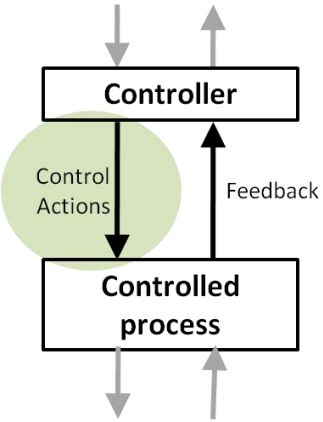
STPA Analysis:
Basic Unsafe Control Action Table

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 Perform ITP when request has been refused	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

Identifying Unsafe Control Actions

Rigorous method

Structure of an Unsafe Control Action



Example:
"Operator provides open train door command when train is moving"

Source (S)

Type (T)

Control Action (CA)

Context (C)

Four parts of an unsafe control action

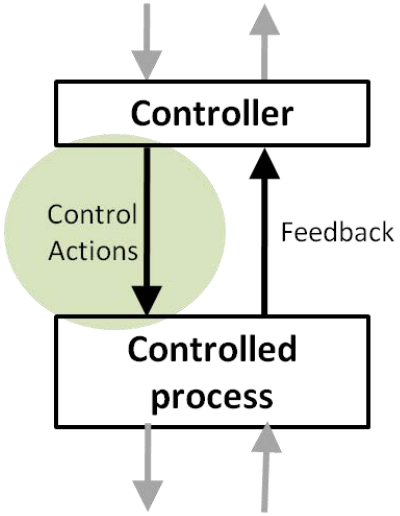
- Source: the controller that can provide the control action
- Type: whether the control action was provided or not provided
- Control Action: the controller's command that was provided / missing
- Context: the system or environmental state in which command is provided

Process Model

Train motion [Stopped / Moving

Train location [At platform / Not Aligned

Rigorous UCA Method



- Identify Unsafe Control Actions
  - Select a Source
  - Select a Control Action
  - Create Process Model
  - Define potential contexts
  - Identify Type 1 UCAs: <source + control action + context>
  - Consider timing
  - Identify Type 2 UCAs: <source + control inaction + context>

Example: Train door controller

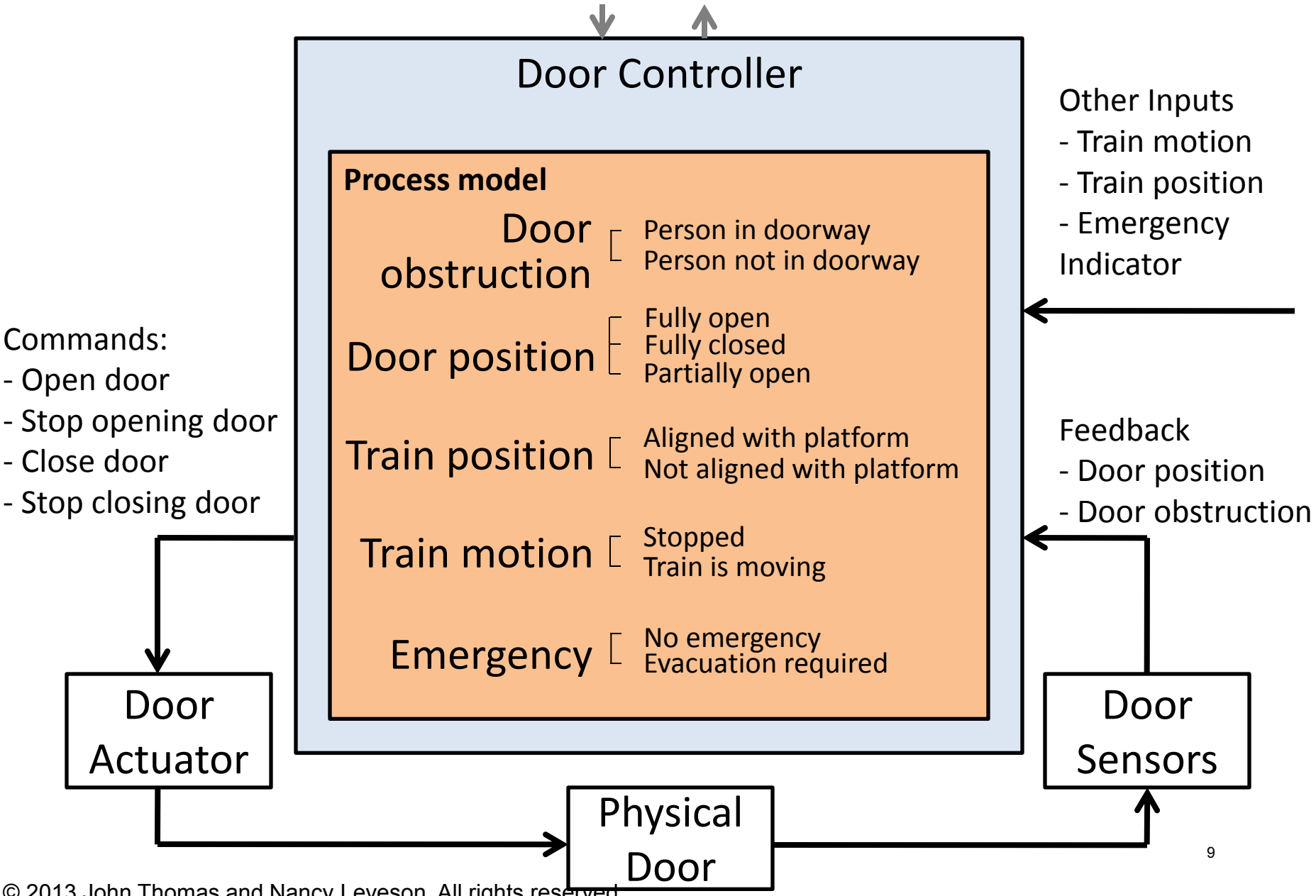
System Hazards

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

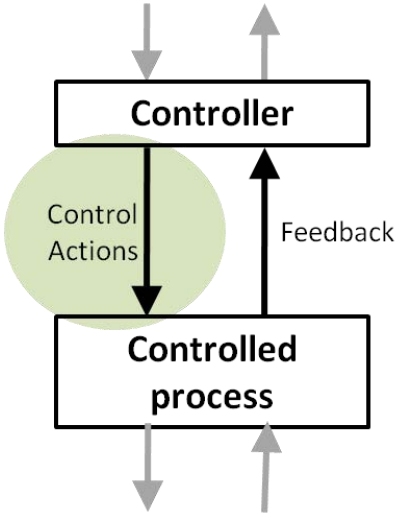
H-2: Doors open when the train is moving or not at platform

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

Example: Control loop



Process



- Identify hazards
- Create control structure
- Identify Unsafe Control Actions
  - Select a Source: Door controller
  - Select a Control Action: Open door command
  - Create Process Model
  - Define potential contexts
    - Identify Type 1 UCAs: <source + control action + context>
    - Consider timing
    - Identify Type 2 UCAs: <source + control inaction + context>

Type 1: Control action *provided*

- Source + Control action
  - *Controller provides door open command*
- Define controller's process model
- Define potential contexts (combinations of process model values)

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
…	…	…	…	…	…

Type 1: Control action *provided*

- Source + Control action
  - *Controller provides door open command*
- Define controller's process model
- Define potential contexts (combinations of process model values)
- Identify Type 1 UCAs: <source + control action + 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**

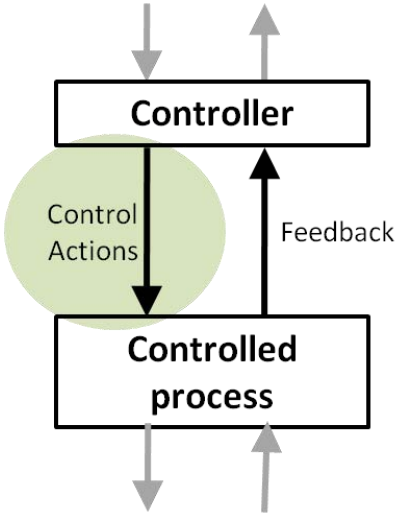
***Design decision: In this situation, evacuate passengers to other cars. Meanwhile, stop the train and then open doors.**

Type 1: Control action *provided*

- Source + Control action
  - *Controller provides door open command*
- Define controller's process model
- Define potential contexts (combinations of process model values)
- Identify Type 1 UCAs: <source + control action + context>
- Consider timing

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**

Process



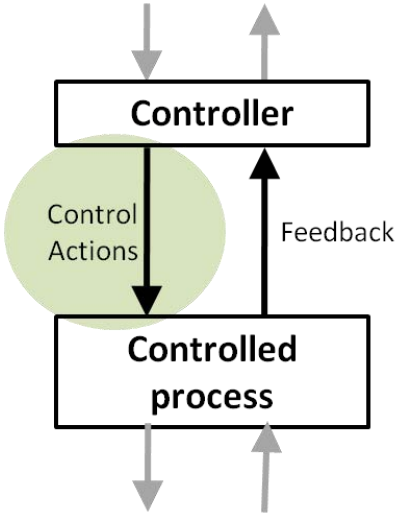
- Identify hazards
- Create control structure
- Identify Unsafe Control Actions
  - Select a Source: Door controller
  - Select a Control Action: Open door command
  - Create Process Model
  - Define potential contexts
  - Identify Type 1 UCAs: <source + control action + context>
  - Consider timing
  - Identify Type 2 UCAs: <source + control inaction + context>

Type 2: Control action *not* provided

- Source + Control action
  - *Controller provides door open command*
- Define controller's process model
- Define potential contexts (combinations of process model values)
- Identify Type 1 UCAs: <source + control action + context>
- Consider timing
- Identify Type 2 UCAs: <source + control inaction + 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**

Process



- Identify hazards
- Create control structure
- Identify Unsafe Control Actions
  - Select a Source: Door controller
  - Select a Control Action: Open door command
  - Create Process Model
  - Define potential contexts
  - Identify Type 1 UCAs: <source + control action + context>
  - Consider timing
  - Identify Type 2 UCAs: <source + control inaction + context>

Resulting List of Unsafe Control Actions

Unsafe Control Actions
UCA 1: Door open command provided while train is moving and there is no emergency
UCA 2: Door open command provided too late while train is stopped and emergency exists
UCA 3: Door open command provided while train is stopped, no emergency, and not at platform
UCA 4: Door open command provided while train is moving and emergency exists
UCA 5: Door open command not provided while train is stopped and emergency exists
UCA 6: Door open command not provided while doors are closing on someone

Parts of this can be automated!

Conversion to Safety Constraints

Unsafe Control Actions	Safety Constraints
UCA 1: Door open command provided while train is moving and there is no emergency	**SC 1:** Door must not be opened while train is moving and there is no emergency
UCA 2: Door open command provided too late while train is stopped and emergency exists	**SC 2:** Door must not be opened while train is stopped and emergency exists
UCA 3: Door open command provided while train is stopped, no emergency, and not at platform	**SC 3:** Door must not be opened while train is stopped, no emergency, and not at platform
UCA 4: Door open command provided while train is moving and emergency exists	**SC 4:** Door must not be opened while train is moving and emergency exists
UCA 5: Door open command not provided while train is stopped and emergency exists	**SC 5:** Door must be opened while train is stopped and emergency exists
UCA 6: Door open command not provided while doors are closing on someone	**SC 6:** Door must be opened while doors are closing on someone

STPA Exercise

a new in-trail procedure
for trans-oceanic flights

Accident (Loss): Two aircraft collide

Hazard: Two aircraft violate minimum separation

STPA Analysis

- More complex control structure

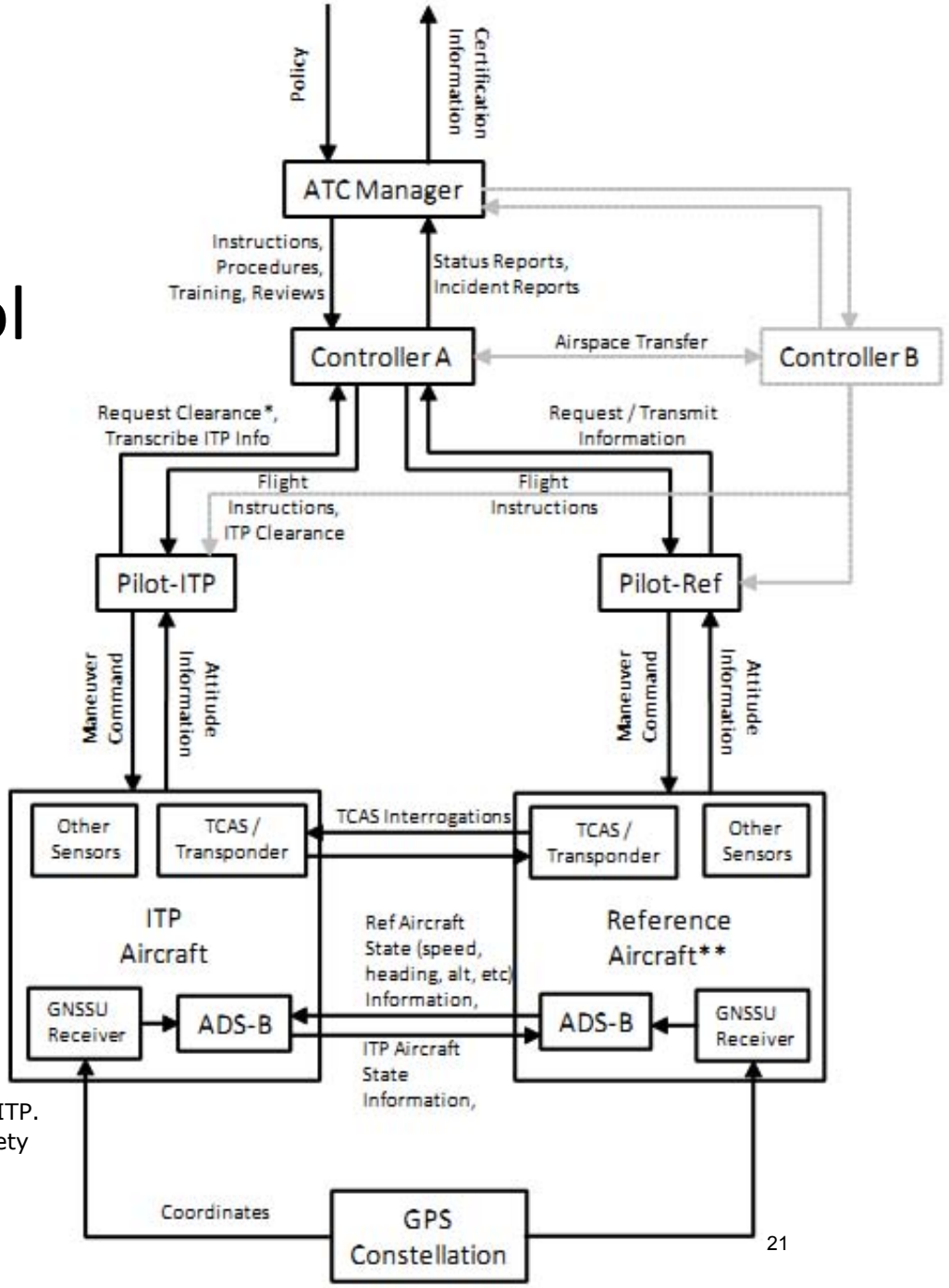


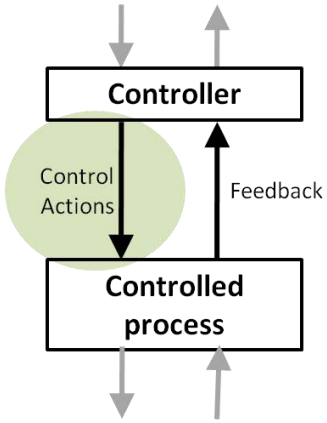
Image: Public Domain. Figure 7: Safety Control Structure for ATSA-ITP. Fleming, Cody Harrison, Melissa Spencer, Nancy Leveson et al. "Safety Assurance in NextGen." March 2012. NASA/CR–2012-217553.

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 Perform ITP when request has been refused	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

Apply rigorous method…

Structure of an Unsafe Control Action



"_____ provides _____ command when _____"

Source — Type — Control Action — Context

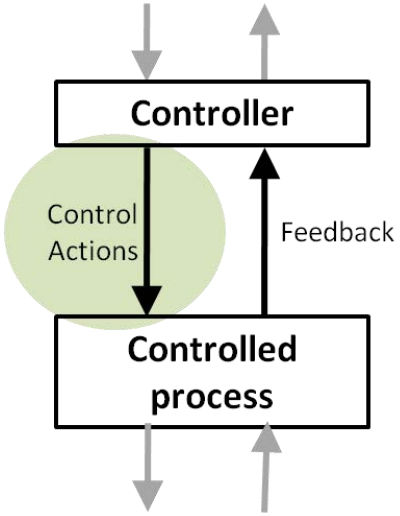
- Source?
  - Pilot
- Control Action?
  - Execute Maneuver
- Context?
  - <create process model>

Process Model

Airspace clear? [Criteria met / Criteria not met

Request status [Approved / Denied

Process



- Identify hazards
- Create control structure
- Identify Unsafe Control Actions
  - Select a Source
  - Select a Control Action
  - Create Process Model
  - Define potential contexts
    - Identify Type 1 UCAs: <source + control action + context>
    - Consider timing
    - Identify Type 2 UCAs: <source + control inaction + context>

Type 1: Control action *provided*

- Source + Control action
  - *Pilot executes maneuver*
- Define controller's process model
- Define potential contexts (combinations of process model values)

Source + Control Action	Airspace Clear?	Request status
Pilot executes maneuver	Criteria met	Request approved
Pilot executes maneuver	Criteria met	Request denied
Pilot executes maneuver	Criteria met	Request not approved or denied
Pilot executes maneuver	Criteria not met	Request approved
Pilot executes maneuver	Criteria not met	Request denied
Pilot executes maneuver	Criteria not met	Request not approved or denied

Type 1: Control action *provided*

- Source + Control action
  - *Pilot executes maneuver*
- Define controller's process model
- Define potential contexts (combinations of process model values)
- Identify Type 1 UCAs: <source + control action + context>

Source + Control Action	Airspace clear?	Request status	Hazardous?
Pilot executes maneuver when…	Criteria met	Request approved	No
	Criteria met	Request denied	Yes
	Criteria met	Request not approved or denied	Yes
	Criteria not met	Request approved	Yes
	Criteria not met	Request denied	Yes
	Criteria not met	Request not approved or denied	Yes

Type 1: Control action *provided*

- Source + Control action
  - *Pilot executes maneuver*
- Define controller's process model
- Define potential contexts (combinations of process model values)
- Identify Type 1 UCAs: <source + control action + context>
- Consider timing

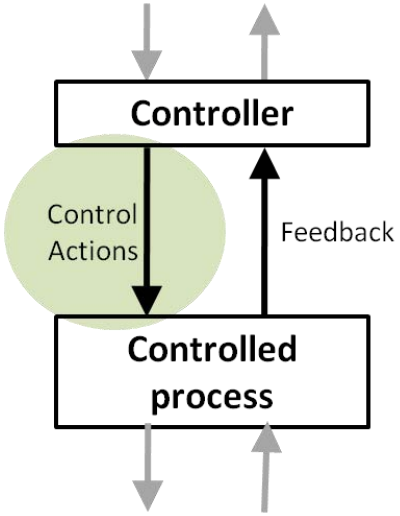
Source + Control Action	Airspace clear?	Request status	Hazardous?	Hazardous if provided too early?	Hazardous if provided too late?
Pilot executes maneuver when…	Criteria met	Request approved	No	No	Yes
	Criteria met	Request denied	Yes	Yes	Yes
	Criteria met	Request not approved or denied	Yes	Yes	Yes
	Criteria not met	Request approved	Yes	Yes	Yes
	Criteria not met	Request denied	Yes	Yes	Yes
	Criteria not met	Request not approved or denied	Yes	Yes	Yes

Type 1: Control action *provided*

Source + Control Action	Airspace clear?	Request status	Hazardous?	Hazardous if provided too early?	Hazardous if provided too late?
Pilot executes maneuver when…	Criteria met	Request approved	No	No	Yes
	Criteria met	Request not approved	Yes	Yes	Yes
	Criteria not met	(doesn't matter)	Yes	Yes	Yes

Table can be simplified

Process



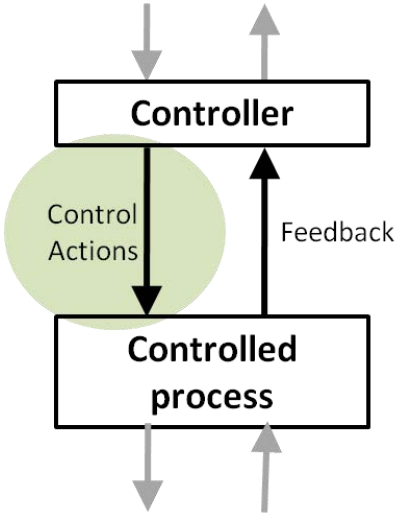
- Identify hazards
- Create control structure
- Identify Unsafe Control Actions
  - Select a Source
  - Select a Control Action
  - Create Process Model
  - Define potential contexts
  - Identify Type 1 UCAs: <source + control action + context>
  - Consider timing
  - Identify Type 2 UCAs: <source + control inaction + context>

Part 2: Control action is *not* provided

- Source + Control action
  - *Pilot executes maneuver*
- Define controller's process model
- Define potential contexts (combinations of process model values)
- Identify Type 1 UCAs: <source + control action + context>
- Consider timing
- Identify Type 2 UCAs: <source + control inaction + context>

Control Action	Request status	Hazardous?
Pilot does not execute ITP when…	Request approved	**Yes**
Pilot does not execute ITP when…	Request denied	**No**
Pilot does not execute ITP when…	Request not approved or denied	**No**

Process



- Identify hazards
- Create control structure
- Identify Unsafe Control Actions
  - Select a Source
  - Select a Control Action
  - Create Process Model
  - Define potential contexts
  - Identify Type 1 UCAs: <source + control action + context>
  - Consider timing
  - Identify Type 2 UCAs: <source + control inaction + context>

STPA Step 2

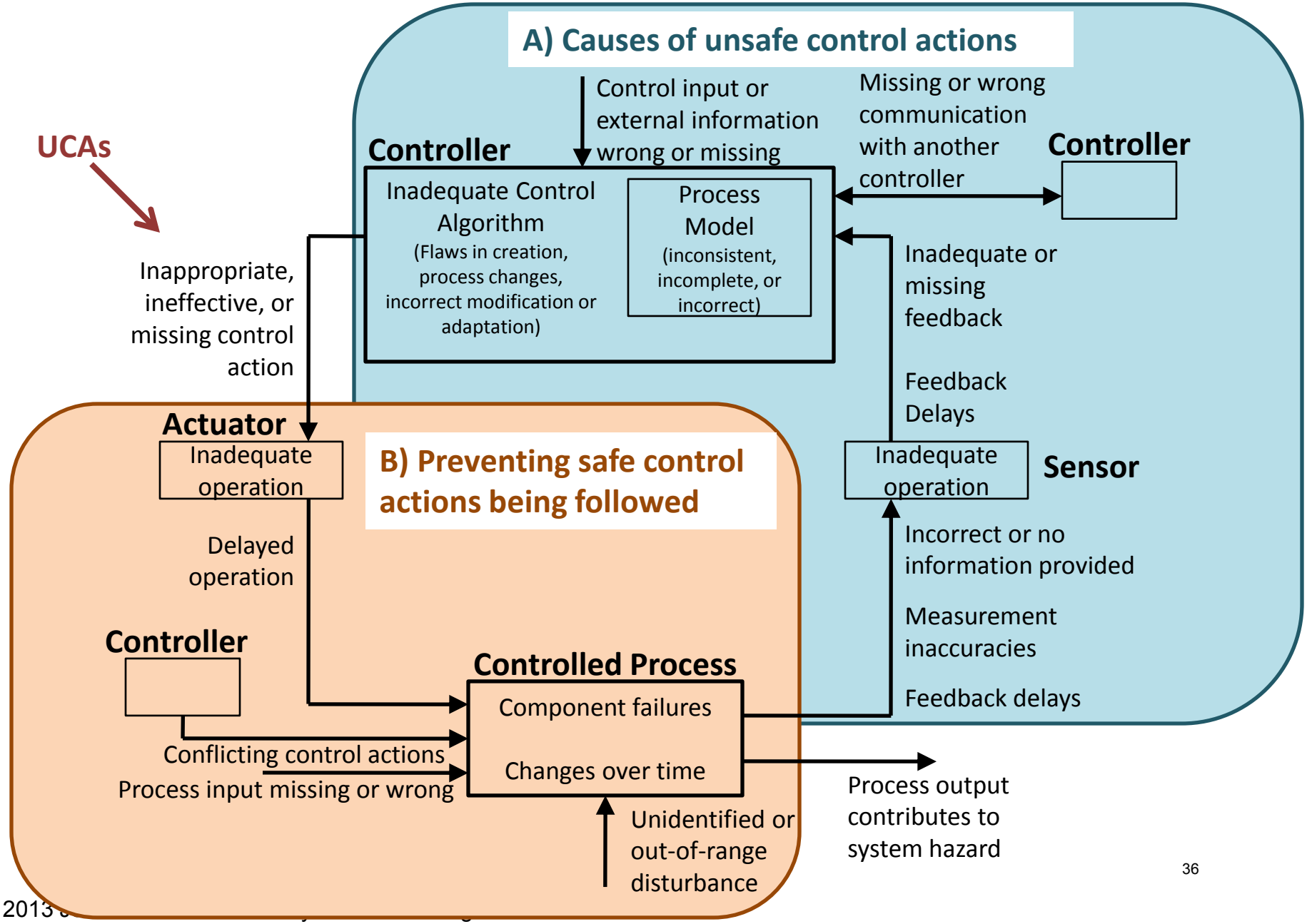
STPA Exercise

- Identify Hazards
- Draw the control structure
  - Identify major components and controllers
  - Label the control/feedback arrows
- Step 1: Identify Unsafe Control Actions (UCAs)
  - Control Table:
    Not given, Given incorrectly, Wrong timing,
    Stopped too soon
  - Create corresponding safety constraints
- Step 2: Identify causal factors
  - Identify controller process models
  - Analyze controller, control path, feedback path, process

STPA Step 2

- Identify causal factors that violate safety constraints
  A. Factors that cause unsafe control actions
  B. Factors the prevent safe control actions being followed

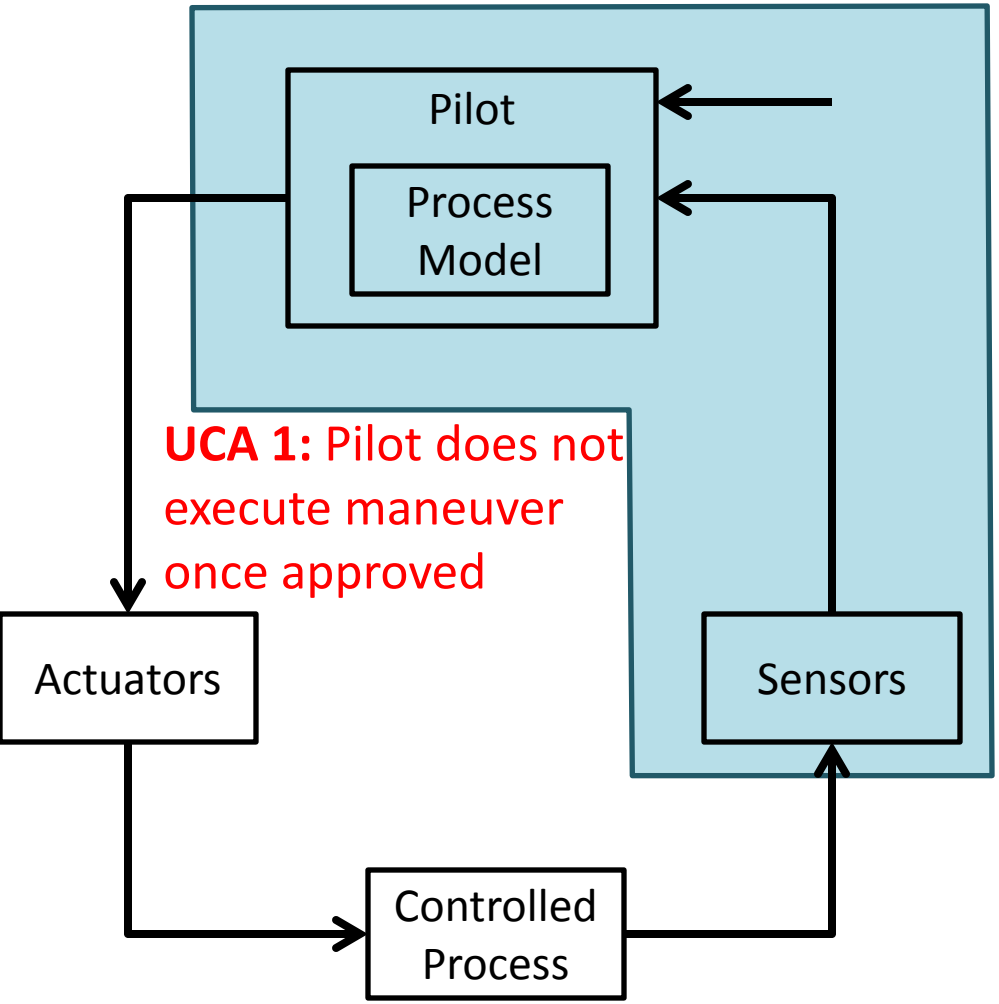
STPA Step 2: Identify Control Flaws



STPA Step 1 output
ITP Example

Unsafe Control Action	Safety Constraint
UCA 1: Pilot does not execute maneuver once it is approved	**SC 1:** Maneuver must be executed once it is approved
UCA 2: Pilot performs ITP when ITP criteria are not met	**SC 2:** Maneuver must not be performed when criteria are not met
UCA 3: Pilot starts maneuver late after having re-verified ITP criteria	**SC 3:** Maneuver must be started within X minutes of re-verifying ITP criteria

STPA 2a: Causes of unsafe control actions



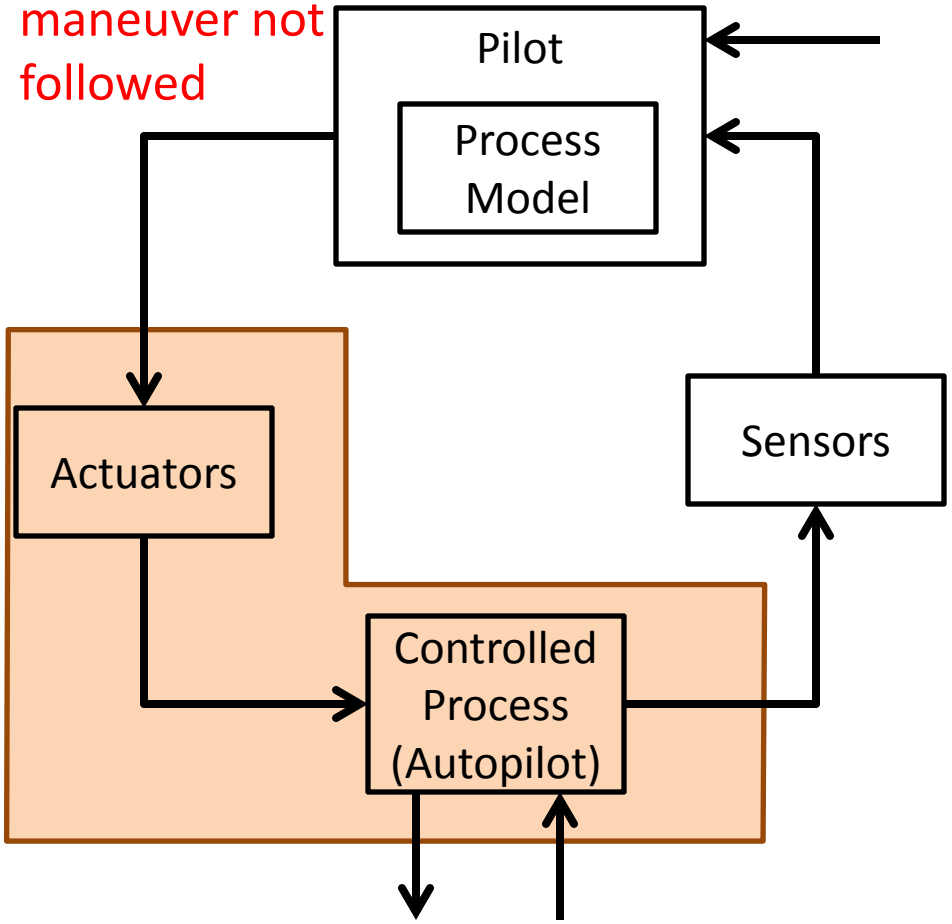
- How could this UCA be caused by:
  - Process model
    - Pilot believes request was denied
    - Pilot believes request was not approved or denied
    - Pilot believes another aircraft is blocking
    - Pilot unsure if another aircraft is blocking
  - Feedback path
    - Equipment shows other traffic in the area
    - Transmission from nearby aircraft received
    - Equipment failure
  - Other inputs
    - Approval not received
    - Rejection received instead of approval
  - Etc.

STPA Step 1 output
ITP Example

Unsafe Control Action	Safety Constraint
UCA 1: Pilot does not execute maneuver once it is approved	**SC 1:** Maneuver must be executed once it is approved
UCA 2: Pilot performs ITP when ITP criteria are not met	**SC 2:** Maneuver must not be performed when criteria are not met
UCA 3: Pilot starts maneuver late after having re-verified ITP criteria	**SC 3:** Maneuver must be started within X minutes of re-verifying ITP criteria

STPA 2b: Safe control action not implemented

SC 1 Violated:
Pilot provides ITP maneuver, but maneuver not followed



- Control action not followed:
  - Control Path
    - Equipment failure
    - Actuator does not execute command
    - Control action delayed
  - Controlled process
    - In wrong mode, ignores control action
    - Responds to control action in unsafe way
    - Receives conflicting commands from other controllers, ignores one or both
    - Physical failures
  - Etc.

16.63J / ESD.03J System Safety
Fall 2012