Common Mistakes in STPA and CAST

Dr. John Thomas

March 2019
Revised October 2019

Any questions? Please email: jthomas4@mit.edu
System-Theoretic Process Analysis (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
Losses,
System-level Hazards
Incorrect Losses

• Loss of brake pressure
• Loss of engine RPM
• Loss of pressurizer pressure
• ...

Incorrect System-level Hazards

- Engine Flameout
- Cruise control does not notify driver of oncoming car
- Improper use of cruise control by driver
- Transmission controller reports incorrect gear to driver

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)

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
Control Structure that is too vague

- Air Traffic Control
  - Instructions
  - Feedback
  - Aircraft

- Flight Crew
  - Instructions
  - Feedback

- Aircraft
  - Updates

- Inputs
  - Updates
Better High-level Control Structure

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

- 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)

- Defined control hierarchy
- Driver is included
Control hierarchy?
Is the network really the ultimate controlled process
No commands to the Robot?
What are the commands/feedback?

Properly defined control hierarchy
Controlled process is the facility/fire
Network is a “pass-through”, not generating its own control actions
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

System-Theoretic Process Analysis (STPA)
Incorrect UCAs (Unsafe Control Actions)

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

“Fails”
“Recognize”
“Monitor”
Missing action
Missing context

Better UCA

- UCA-1:
  Pilot does not provide pitch up cmd when TCAS provides climb TA [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.
Incorrectly interpreting UCA

Controller does not provide Go cmd when obstacle is in path

Controller provides Go cmd when obstacle is in path

Controller does not provide Stop cmd when obstacle is in path

Controller provides Stop cmd when obstacle is in path

Don’t assume the Go cmd must have been provided. Maybe, maybe not. It could also be that no cmd was provided.

“It must have provided the Go command”
## Incomplete UCAs

**Diagram:**
- **Controller**
  - Stop (0 speed)
  - Go (speed != 0)
- **Actuator**
- **Sensor**
- **Controlled Process**

**Table:**

<table>
<thead>
<tr>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early, too late, out of order</th>
<th>Applied too long, Stopped too soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller does not provide Go cmd when _________</td>
<td>Controller provides Go cmd when obstacle is in path</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Controller does not provide Stop cmd when obstacle is in path</td>
<td>Controller provides Stop cmd when _________</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Other UCAs are missing. What about:**
- ... 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) ...
“This research found that STPA was weaker on system failures: [link]”

A UCA contains five parts:

**UCA-2: BSCU Autobrake provides Brake command during a normal takeoff [H-4.3]**

<table>
<thead>
<tr>
<th>No.</th>
<th>Command or Event</th>
<th>Not Provided</th>
<th>Provided Unsafe</th>
<th>Provided Too Early</th>
<th>Provided Too Late</th>
<th>Out of Sequence</th>
<th>Stopped Too Soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vehicle Status Signal</td>
<td>Catastrophic- (Wrong brake pressure determination) [1a]</td>
<td>Catastrophic- (Wrong brake pressure determination) [1a]</td>
<td>N/A</td>
<td>Catastrophic- (Wrong brake pressure determination and wrong reaction time) [1a]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>Object Status Signal</td>
<td>Catastrophic- (Wrong brake pressure determination) [2a]</td>
<td>Catastrophic- (Wrong brake pressure determination) [2a]</td>
<td>N/A</td>
<td>Catastrophic- (Wrong brake pressure determination and wrong reaction time) [2a]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Table 1. Inadequate Control Commands/Events*
“This research found that STPA was weaker on system failures: [link]”

HAZARD ANALYSIS

For hazard analysis the detailed control structure diagram of the system was acquired. Next, the first and the second author of this study analyzed the forward collision avoidance system and identified 14 inadequate control commands or events, including their causal factors. The results (both inadequate control commands or events and their causal factors) were analyzed and reviewed by the third and the fourth author. In this study, the authors have performed hazard analysis of the forward collision avoidance system by following their best interpretation/understanding of the STPA guidelines as presented by Leveson (2012) and Leveson et al. (2012). Table 1 shows an excerpt of the identified inadequate control commands or events\(^1\) that could lead to hazardous states.

<table>
<thead>
<tr>
<th>No.</th>
<th>Command or Event</th>
<th>Not Provided</th>
<th>Provided Unsafe</th>
<th>Provided</th>
<th>Out of Sequence</th>
<th>Stopped Too Soon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Too Early</td>
<td>Too Late</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Vehicle Status Signal</td>
<td>Catastrophic- (Wrong brake pressure determination) [1a]</td>
<td>Catastrophic- (Wrong brake pressure determination) [1a]</td>
<td>N/A</td>
<td>Catastrophic- (Wrong brake pressure determination and wrong reaction time) [1a]</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>Object Status Signal</td>
<td>Catastrophic- (Wrong brake pressure determination) [2a]</td>
<td>Catastrophic- (Wrong brake pressure determination) [2a]</td>
<td>N/A</td>
<td>Catastrophic- (Wrong brake pressure determination and wrong reaction time) [2a]</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 1. Inadequate Control Commands/Events

- STPA Steps 1 & 2?
- Incorrect STPA Step 3
- STPA Step 4?
Conclusions despite mistakes

• "STPA has proved to be an effective and efficient hazard analysis method“

• "With regard to software error type hazards, STPA found more hazards than FMEA of unique hazards“

• "STPA considers more types of hazard causes than the other traditional hazard analysis methods. Therefore, STPA is more complete than existing traditional hazard analysis methods"
<table>
<thead>
<tr>
<th>Command /event</th>
<th>Not provided</th>
<th>Provided unsafe</th>
<th>Provided ...</th>
<th>Stopped too soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object detection signal</td>
<td>Catastrophic-system dysfunction [collision] (1a)</td>
<td>Catastrophic-system malfunctioning (1b)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Vehicle complex signal</td>
<td>Catastrophic-problem in calculation of vehicle status and collision probability (2a)</td>
<td>Catastrophic-problem in calculation of vehicle status and collision probability (2a)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Collision warning signal</td>
<td>Negligible (if every thing is working properly, then the active safety will be saved from collision) (3a)</td>
<td>Negligible (if every thing is working properly, then the active safety will be saved from collision) (3a)</td>
<td>N/A</td>
<td>Negligible (warning will be stopped too soon that can cause accident. If everything works properly, then the active safety will be saved from collision) (3b)</td>
</tr>
</tbody>
</table>

Incorrect unsafe control actions
Incorrect UCAs

Billing System (singular)

Process Model:
- For each Meter:
  - Tracks Registered OR Retired state
  - Collection of Meter readings and date sent
  - If Bill exists for Meter
  - If Bill has been paid
  - Current Week counter

Responsibilities:
- Track all Meters
- Send commands in response to Meters exceeding allowed leeway in reporting readings or in paying bills.

Feedback:
- Pay bill
- Provide reading

Meter (many)

Responsibilities:
- Report meter readings at frequent intervals
- Pay bills when issued

Feedback:
- Connection status
- Electricity usage/readings

Commands:
- Disconnect from supply

Commands:
- Notify bill has been generated
- Disconnect

Electricity supply (underlying process) [one per meter]

Figure 1: Functional control structure for smart meter example.

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Is issued</th>
<th>Is not issued</th>
<th>Is issued out of sequence</th>
<th>Is issued for incorrect duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Meter</td>
<td>An invalid meter is re-registered.</td>
<td>A meter fails to be registered.</td>
<td>A meter is registered multiple times.</td>
<td>N/A - registration is discrete.</td>
</tr>
</tbody>
</table>

Table 2: Control action analysis results
Incorrect UCAs

- UCA must specify the context that makes the control action unsafe
- What does “erroneously” mean? What makes it unsafe?
Indirect context

- **Controller beliefs belong in another step**
- **Ask: what is the condition that makes the park command itself unsafe?**

### Table

<table>
<thead>
<tr>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early, too late, out of order</th>
<th>Applied too long, Stopped too soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>Driver provides Park when they incorrectly believe vehicle is stopped</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Controller does not provide Stop cmd when</td>
<td>Controller provides Stop cmd when</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>_______</td>
<td>_______</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Vague context, assumptions

<table>
<thead>
<tr>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early, too late, out of order</th>
<th>Applied too long, Stopped too soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator does not provide increase turbine speed cmd when required</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Defining UCAs relative to procedures

STPA does not assume the existing procedure is fully correct and complete. Better UCA:
- Operator provides increase turbine speed cmd when turbine speed exceeds X rpm
Confusing UCAs with Failure Effects

Are these correct? Hard to review. These were reviewed incorrectly.

Tips:
• UCAs are control actions in a context that makes them unsafe
• UCAs are not just effects
• UCAs are not just hazardous states
• UCA contexts might be non-hazardous without the control action.
Confusing UCA contexts with hazardous states

A UCA is an action that is unsafe in some context. Confusion can be avoided by writing whole UCA.

UCA-1: Protection System does not provide Open MSIV Cmd when Steam Generator Tube Ruptures [H-1,2]
Potential confusion

Operator

Close MSIV Cmd

Close MSIV Cmd

MSIV (Main Steam Isolation Valve)

Not providing causes hazard | Providing causes hazard | Too early, too late, out of order | Applied too long, Stopped too soon
---|---|---|---
when SGTR and other cooling systems not operational | ... | ... | ...
Potential confusion

<table>
<thead>
<tr>
<th>Close MSIV Cmd</th>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early, too late, out of order</th>
<th>Applied too long, Stopped too soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close MSIV Cmd</td>
<td>when SGTR and other cooling systems operational</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Confusing control actions from multiple controllers

Operator

- Close MSIV Cmd

Protection System

- Close MSIV Cmd

Diverse Actuation System

- Close MSIV Cmd

MSIV (Main Steam Isolation Valve)

<table>
<thead>
<tr>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early, too late, out of order</th>
<th>Applied too long, Stopped too soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>when SGTR and other cooling systems operational</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Confusion can be avoided by writing whole UCA.

Author identified a valid UCA, but it was not inadequately communicated to reviewers and others.

UCA-1: Operator does not provide Close MSIV Cmd when SGTR and other systems operational [H-1,2]
Confusing control actions from multiple controllers

- Confusion can be avoided by writing whole UCA.

Operator

<table>
<thead>
<tr>
<th></th>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early, too late, out of order</th>
<th>Applied too long, Stopped too soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brake Cmd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>when collision is imminent</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Adaptive Cruise Control

Automatic Emergency Braking

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

UCA-2: Adaptive Cruise Control does not provide Brake Cmd when distance to leading vehicle is below setpoint [H-1,2,3]

Author identified a valid UCA, but it was not inadequately communicated to reviewers and others.
Current guidance

**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)

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
Identifying causal factors without interactions
Causal factors should be more than failures and malfunctions

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

**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 a 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 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 Scenario Example

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

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

Visual feedback doesn’t clearly indicate HTV motion

Design does not indicate the measured distance to Crew

© Copyright John Thomas 2018
Better Scenario Example

Driver accelerates when vehicle is not in appropriate range (e.g. reverse instead of drive)

Driver incorrectly believes vehicle is in Drive

MM not updated because vehicle ignored cmd to shift to Drive (stayed in reverse)