STPA Step 4
Building Formal Scenarios

A New Scenario Approach

John Thomas
Tutorial Objective

- These short tutorials are **not training classes**
- We cannot cover everything in these tutorial sessions.
- The objective is to introduce some of the core concepts so new attendees can follow the presentations this week.
- Training and practice with a qualified instructor are needed to apply these techniques and become proficient (as with most techniques). These short tutorials are subsets of larger training classes.

Any questions? Email me! [JThomas4@mit.edu](mailto:JThomas4@mit.edu)
1) Define Purpose of the Analysis
2) Model the Control Structure
3) Identify Unsafe Control Actions
4) Identify Loss Scenarios

(Leveson and Thomas, 2018)
New STPA scenario approach has been tested for 8 years

Examples:

- **(Nuclear)** A New Process for Building STPA Causal Scenarios, John Thomas, 2016 MIT STAMP Workshop
- **(Space)** A Process for STPA: STAMP Accident Model of HITOMI and Expansion to Future Safety Culture, John Thomas and Nancy Leveson (MIT), Masa Katahira and Naoki Ishimama (JAXA), Nobuyuki Hoshino (JAMSS), 2017 MIT STAMP Workshop
- **(Auto)** STPA Applied to Autonomous Vehicles, Jeff Stafford (Renesas), John Thomas (MIT), 2019 MIT STAMP Workshop
- **(Software)** STPA Applied to AV Software, Shaun Mooney (Codethink), John Thomas (MIT), 2019
- **(Communications)** Brittany Bishop, MIT Thesis, 2024
- **(Software)** Qualcomm, 2023-2024
- **(Software)** Google, 2023-2024

In all cases, the new approach has found loss scenarios and causes that had been previously overlooked.
# New Scenario Approach

## Advantages and Disadvantages Observed

<table>
<thead>
<tr>
<th>Disadvantages</th>
<th>Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>• More rules and structure</td>
<td>• The rules and structure provide more guidance</td>
</tr>
<tr>
<td>• Takes longer to teach &amp; learn</td>
<td>• Enables a more directed search, less ad-hoc and informal</td>
</tr>
<tr>
<td>• Unclear if it takes longer to perform (less time in some cases)</td>
<td>• The rationale for the scenarios and how you found them is clearer.</td>
</tr>
<tr>
<td></td>
<td>• So far, the new process has always captured additional cases that were previously overlooked</td>
</tr>
<tr>
<td></td>
<td>• The top-down approach was more scalable to extremely complex systems compared to previous STPA attempts</td>
</tr>
<tr>
<td></td>
<td>• Less repetition in the results (shorter documentation, higher information density)</td>
</tr>
<tr>
<td></td>
<td>• Provides clear exit criteria to rigorously review and find gaps</td>
</tr>
<tr>
<td></td>
<td>• Enables automation of some parts of Step 4</td>
</tr>
<tr>
<td></td>
<td>• The formal structure can enable mathematical proofs for properties like scenario coverage</td>
</tr>
<tr>
<td></td>
<td>• Improved consistency and repeatability. Less dependence on who is doing the analysis.</td>
</tr>
</tbody>
</table>
Goals for new scenario approach

• Provide scenario guidance for new practitioners who may get stuck
• Provide a formal structure for scenarios (similar to UCA syntax)
• Provide a way to review scenario completeness and find gaps
• Handle more complexity: Implement a top-down strategy, not a backward search strategy.
Example UCA: “UCA-1: Computer provides Shift-to-Park cmd while vehicle is moving [H-1]”

Can we make STPA Step 4 more like this?
Forward vs. Backward Search

Initiating Events | Final States
---|---
A | W: nonhazard
B | X: HAZARD
C | Y: nonhazard
D | Z: nonhazard

Initiating Events | Final States
---|---
A | W: nonhazard
B | X: HAZARD
C | Y: nonhazard
D | Z: nonhazard

Forward Search

Backward Search
Top-Down Search

TOP EVENT (Hazard)

Intermediate or pseudo-events

Basic or primary events
Building Scenarios

1. Define small number of high-level scenarios
   – Start with broad, abstract scenarios
   – Consider each class of scenario factors
   – Easy to review, show coverage, completeness, etc.

2. Identify potential solutions
   – Requirements
   – Modify control actions
   – Modify types of feedback
   – Modify responsibilities
   – Etc.

3. Refine into more detailed scenarios (if solutions not found)
   – Include more design detail
   – Can be done in parallel with development
STPA Step 4A: Identify scenarios that cause UCAs

Unsafe Control Action (UCA)

Controller
- Decision Making (Flaws in creation, process changes, incorrect modification or adaptation)
- Process Model (inconsistent, incomplete, or incorrect)
- Control input or external information wrong or missing
- Missing or wrong communication with another controller
- Inadequate or missing feedback

Actuator
- Inadequate operation
- Delays, inaccuracies, missing/incorrect behavior

Sensor
- Inadequate operation
- Incorrect or no information provided
- Measurement inaccuracies
- Feedback delays

Controlled Process
- Component failures
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard

Controller
- Conflicting control actions
- Process input missing or wrong
STPA Step 4A: Identify scenarios that cause UCAs

Unsafe Control Action (UCA)

Controller
- Decision Making (Flaws in creation, process changes, incorrect modification or adaptation)
- Process Model (inconsistent, incomplete, or incorrect)
- Control input or external information wrong or missing
- Missing or wrong communication with another controller
- Inadequate or missing feedback
- Feedback Delays

Actuator
- Inadequate operation
- Delays, inaccuracies, missing/incorrect behavior

Sensor
- Inadequate operation
- Incorrect or no information provided
- Measurement inaccuracies
- Feedback delays

Controller
- Conflicting control actions
- Process input missing or wrong

Controlled Process
- Component failures
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard

Why?
STPA Step 4B: Potential control actions not followed properly

Control Action or Inaction

Decision Making
(Flaws in creation, process changes, incorrect modification or adaptation)

Controller

Control input or external information wrong or missing

Process Model
(inconsistent, incomplete, or incorrect)

Controller

Missing or wrong communication with another controller

Inadequate or missing feedback

Feedback Delays

Inadequate operation

Sensor

Incorrect or no information provided

Measurement inaccuracies

Controller

Conflicting control actions

Component failures

Controller

Changes over time

Process input missing or wrong

Not Executed Properly

Controller

Changes over time

Unidentified or out-of-range disturbance

Controller

Sensor

Controller

Actuator

Inadequate operation

Delays, inaccuracies, missing/incorrect behavior

Controller

Process output contributes to system hazard

Incorrect or no information provided

Measurement inaccuracies

Controller

Missing or wrong communication with another controller

Controller

Controller

STPA Step 4B: Potential control actions not followed properly

Component failures

Controller

Changes over time

Unidentified or out-of-range disturbance

© Copyright John Thomas 2017
Four Classes of Formal Scenarios

Class 1
- Controller
- Feedback or other inputs
- GOOD
- UNSAFE
- Control Actions
- Controlled Process

Class 2
- Controller
- Feedback or other inputs
- UNSAFE
- Control Actions
- Controlled Process

Class 3
- Controller
- Feedback or other inputs
- GOOD
- Control Actions
- Controlled Process

Class 4
- Controller
- Feedback or other inputs
- Control Actions
- Controlled Process

© Copyright 2024 John Thomas
STPA Step 4A: Identify scenarios that cause UCAs

UCA-2:  
<Controller> provides <Control Action> when <Context>

Controller
- Decision Making (Flaws in creation, process changes, incorrect modification or adaptation)
- Process Model (inconsistent, incomplete, or incorrect)
- Control input or external information wrong or missing
- Missing or wrong communication with another controller

Actuator
- Inadequate operation
- Delays, inaccuracies, missing/incorrect behavior
- Conflicting control actions
- Process input missing or wrong

Sensor
- Inadequate operation
- Incorrect or no information provided
- Measurement inaccuracies
- Feedback delays

Controlled Process
- Component failures
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard

Class 1:
<Feedback/input> ___ was adequate

Class 2:
<Feedback/input> ___ was inadequate

© Copyright 2024 John Thomas
UCA-2: <Controller> provides <Control Action> when <Context>.

Decision Making (Flaws in creation, process changes, incorrect modification or adaptation)

Process Model (inconsistent, incomplete, or incorrect)

Controller input or external information wrong or missing

Missing or wrong communication with another controller

Inadequate or missing feedback

Feedback Delays

Component failures

Unidentified or out-of-range disturbance

Changes over time

Process output contributes to system hazard

Incorrect or no information provided

Measurement inaccuracies

Feedback delays

Controller

Actuator

Inadequate operation

Delays, inaccuracies, missing/incorrect behavior

Controller

Sensor

Inadequate operation

Inadequate or missing feedback

Feedback Delays

Controller

STPA Step 4A: Identify scenarios that cause UCAs

Class 1: <Input> correctly showed that...
STPA Step 4A: Identify scenarios that cause UCAs

Class 1 Scenario Archetype:
- **Output**: UCA-1: <Controller> provides <Control Action> when <Context>
- **Input**: Class 1: <Input> correctly showed that <Context>
General Transfer Function Concept

Class 1 Scenario Archetype:
- **Output**: UCA-1: <Controller> provides <Control Action> when <Context>
- **Input**: Class 1: <Input> correctly showed that <Context>
Controller

One Output is Fixed

Ask: Why would <controller> make this decision?

One Input is Fixed
UCA-2: `<Controller>` provides `<Control Action>` when `<Context>`

Controller -> Decision Making (Flaws in creation, process changes, incorrect modification or adaptation)

Controller -> Process Model (inconsistent, incomplete, or incorrect)

Control input or external information wrong or missing
Missing or wrong communication with another controller

Class 1: `<Input>` correctly showed that `<Context>`

Why would `<Controller>` make this decision?

Consider:
1) Failure causes
2) No failures
STPA Step 4A: Identify scenarios that cause UCAs

UCA-2: <Controller> provides <Control Action> when <Context>

Decision Making
(Flaws in creation, process changes, incorrect modification or adaptation)

Process Model
(inconsistent, incomplete, or incorrect)

Why would <Controller> make this decision?

• Because <Controller> is designed to default to ____ if _____ (will ignore ____ and ____)
• Because <Controller> also received ____, so it appeared reasonable to <Control Action>
• Because there is no other _____, so other control actions were not feasible
• Etc.
Scenario Archetypes

Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

© Copyright 2024 John Thomas
Example: Mars Polar Lander

![Mars Polar Lander illustration]

**Thruster Controller**

- Control Algorithm
- Process Model (beliefs)

- Disable Thruster
- Touchdown

**Lander**

John Thomas, 2019
Mars Polar Lander

Thruster Controller

Control Algorithm

Process Model (beliefs)

Disable Thruster

Touchdown

Lander

Unsafe Control Action (UCA): Computer provides Disable-Thruster cmd when spacecraft is in the air

Process Model: Incorrectly believes spacecraft is on ground

Feedback: Touchdown indication received when in air

Physical Interaction: Simultaneous leg vibration during leg deployment
UCA-2: Thruster Controller provides Disable-Thruster Cmd when spacecraft is in the air.

Scenario Archetypes

Class 1 Scenario Archetype: Inadequate Controller Behavior
- UCA: <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____
UCA-2: Thruster Controller provides Disable-Thruster Cmd when spacecraft is in the air

Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- <Input> to <Controller> correctly indicated _____

Scenario Archetypes
UCA-2: Thruster Controller provides Disable-Thruster Cmd when spacecraft is in the air

Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it's in the air
UCA-2: <Controller> provides ____ when ____

Scenario Archetypes

Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

Ask: Why would <Controller> make this decision?
UCA-2: 
(Controller) provides ____ when ____

Class 1 Scenario Archetype: Inadequate Controller Behavior
- (Controller) provides _____ when _____
- (Input) to (Controller) correctly indicated ______

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Why would <Controller> make this decision?

- Because <Controller> is designed to default to ____ if _____ (will ignore ____ and ____)
- Because <Controller> also received ____, so it appeared reasonable to <Control Action>
- Because there is no other _____, so other control actions were not feasible
- Etc.
If you have trouble, a generic controller model can help you ask better questions & find actionable causes.
Generic Controller Model

Controller

Control Actions

Inputs/Feedback

**Class 1 Scenario Archetype:**

**Output:** UCA

**Input:** Correctly indicates UCA context (but other inputs may override or conflict)

© Copyright 2024 John Thomas
Generic Controller Model

Controller

Decision Making (Control Algorithm)

Process Model (Beliefs/States)

Interpretation (update process model)

Responsibilities

Current state

Control Actions

Other Information

Feedback

Class 1 Scenario Archetype:

Output: UCA

Input: Correctly indicates UCA context (but other inputs may override or conflict)

© Copyright 2024 John Thomas
Class 1 Scenario Archetype:

Output: UCA

Input: Correctly indicates UCA context (but other inputs may override or conflict)

Why would <controller> make this decision?
Refining Class 1 Scenario Archetype: Inadequate Controller Behavior

Common causes of Scenario Archetype 1:

A. Responsibilities (e.g., desired end states) that would produce this UCA
B. Control algorithms or decision-making rationale that would explain the UCA
C. Process models that would explain this UCA
D. Interpretation rules / process model updates that would explain the UCA
E. Internal controller states/modes that would explain the UCA (failure, lame duck mode, etc.)
F. Controller inputs (control actions, feedback, or other inputs) that would explain the UCA
   - E.g., Conflicting/contradictory inputs, inputs from another controller, etc.
   - If the input is another UCA, then make sure the new UCA is recorded in STPA Step 3. The new UCA will be analyzed using the same process.
G. Etc.
UCA-2: 
(Controller) provides _____ when _____

Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
  - <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

Responsibilities
(Controller) by design is responsible for always assigning _____ to every ___ (even if _____.)

Discussion: Responsibilities can be:
- Fixed (hardcoded, embedded in design of algorithm)
- Dynamic (provided in real-time as a control action input into the controller)
- Adaptive (developed and changed by the controller as needed)
Class 1 Scenario Archetype: Inadequate Controller Behavior
- \(<\text{Controller}>\) provides _____ when _____
- \(<\text{Input}>\) to \(<\text{Controller}>\) correctly indicated ______

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Control Algorithms
\(<\text{Controller}>\) control algorithm is designed to fall back onto ____ strategy if _____

© Copyright 2024 John Thomas
Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

Process Models

PM-1: <Controller> incorrectly believes _____

© Copyright 2024 John Thomas
Scenario Archetypes

Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

Interpretation of inputs

In some situations, <Controller> will interpret ____ as an indicator of ___. This interpretation can underestimate ___, causing ___.

When <Feedback/input 1> conflicts with <feedback/input 2>, <Controller> may assume ____

When <feedback/input> is not available, <Controller> may assume ___

UCA-2: <Controller> provides ____ when ____
Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Scenario Archetypes

Refined Scenarios

Update Process Model

<table>
<thead>
<tr>
<th>Unsafe PM Update</th>
<th>Unsafe Feedback or Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM-1 is not updated when _</td>
<td>Feedback/input ___ is (or is not) provided when ___</td>
</tr>
<tr>
<td>PM-1 is the initial (default) belief before feedback/input ___ received</td>
<td></td>
</tr>
<tr>
<td>PM-1 is updated incorrectly due to feedback/input ___ that indicates ___</td>
<td>Feedback/input ___ is provided when ___</td>
</tr>
<tr>
<td>PM-1 is updated too late (or early) due to ___</td>
<td>Feedback/input ___ is delayed (or too early) when ___</td>
</tr>
<tr>
<td>PM-1 stops updating too soon before ___</td>
<td>Feedback/input ___ is applied too long after (stopped too soon before)</td>
</tr>
<tr>
<td>PM-1 continues to be updated too long after ___</td>
<td></td>
</tr>
</tbody>
</table>
Updated Process Model

<table>
<thead>
<tr>
<th>Unsafe PM Update</th>
<th>Unsafe Feedback or Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM-1 is not updated when ___ PM-1 is the initial (default) belief before feedback/input ___ received</td>
<td>Feedback/input ___ is (or is not) provided when ___</td>
</tr>
<tr>
<td>PM-1 is updated incorrectly due to feedback/input ___ that indicates ___</td>
<td>Feedback/input ___ is provided when ___</td>
</tr>
<tr>
<td>PM-1 is updated too late (or early) due to ___</td>
<td>Feedback/input ___ is delayed (or too early) when ___</td>
</tr>
<tr>
<td>PM-1 stops updating too soon before ___ PM-1 continues to be updated too long after ___</td>
<td>Feedback/input ___ is applied too long after (stopped too soon before) ___</td>
</tr>
</tbody>
</table>
Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

Controller States / Modes
- If <Controller> is in ____ mode, it will continue to <Control Action> using alternate input ____.
- If <Controller> ___ is disabled, then <Controller> can _____.

© Copyright 2024 John Thomas
Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Other Inputs

<Controller> does not prevent <Control Action> when alternate input ___ is received.
Scenario Archetypes

Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype:

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

Other inputs

- Although <Input> is correct, the feedback from ____ may be incorrect and may cause <Controller> to ____.
Four Classes of Formal Scenarios

Class 1
- Controller
  - Feedback or other inputs
  - GOOD
- Controlled Process
  - UNSAFE

Class 2
- Controller
  - Feedback or other inputs
  - UNSAFE
- Controlled Process
  - UNSAFE

Class 3
- Controller
  - Feedback or other inputs
  - GOOD
- Controlled Process
  - UNSAFE
- Controlled Process
  - UNSAFE

Class 4
- Controller
  - Feedback or other inputs
- Controlled Process
  - UNSAFE
- Controlled Process
  - GOOD
STPA Step 4A: Identify scenarios that cause UCAs

UCA-2: 
<Controller> provides <Control Action> when <Context>

Controller

Decision Making
(Flaws in creation, process changes, incorrect modification or adaptation)

Process Model
(inconsistent, incomplete, or incorrect)

Control input or external information wrong or missing

Missing or wrong communication with another controller

Inadequate or missing feedback

Feedback Delays

Class 1: Feedback/input ___ was adequate

Class 2: Feedback/input ___ was inadequate

Actuator

Inadequate operation

Inadequate operation

Sensor

Inadequate operation

Incorrect or no information provided

Measurement inaccuracies

Feedback delays

Controller

Conflicting control actions

Feedback Delays

Unidentified or out-of-range disturbance

Changes over time

Process output contributes to system hazard

Process input missing or wrong

Controller

Component failures

Delays, inaccuracies, missing/incorrect behavior
UCA-2: 
<Controller> provides 
<Control Action> when 
<Context>

Decision Making (Flaws in creation, process changes, incorrect modification or adaptation)

Process Model (inconsistent, incomplete, or incorrect)

Control input or external information wrong or missing

Missing or wrong communication with another controller

Inadequate or missing feedback

Feedback Delays

Inadequate operation

Controller

Actuator

Inadequate operation

Delays, inaccuracies, missing/incorrect behavior

Controller

Sensor

Inadequate operation

Incorrect or no information provided

Measurement inaccuracies

Feedback delays

Controller

Controlled Process

Component failures

Changes over time

Conflicting control actions

Process input missing or wrong

Unidentified or out-of-range disturbance

Process output contributes to system hazard

Class 2: Feedback/input ____ was inadequate

STPA Step 4: Class 2 Scenario Archetype
STPA Step 4: Class 2 Scenario Archetype

Class 2 Scenario Archetype:
- Feedback to <Controller> did not adequately indicate <Context>
- <Context> is true

UCA-2: <Controller> provides <Control Action> when <Context>

Class 2: Feedback/Input did not adequately indicate ____

Sensor
- Inadequate operation
- Inadequate or missing feedback
- Feedback Delays
- Incorrect or no information provided
- Measurement inaccuracies
- Feedback delays

Controlled Process
- Component failures
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard

<UCA Context> is true
STPA Step 4: Class 2 Scenario Archetype

UCA-2: 
<Controller> provides <Control Action> when <Context>

Class 2: Feedback/Input did not adequately indicate ____

Why?

Sensor
- Inadequate operation
- Inadequate or missing feedback
- Feedback Delays
- Component failures
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard

Controlled Process
- Component failures
- Changes over time
- Unidentified or out-of-range disturbance

© Copyright 2024 John Thomas
UCA-2: 
<Controller> provides <Control Action> when <Context>
STPA Step 4: Class 2 Scenario Archetype

UCA-2: <Controller> provides <Control Action> when <Context> is true

Why? (non-failures)
- Because the ___metric is estimated from ___ that uses different ___
- Because the ___ feedback is outdated due to ___ second feedback delay
- Etc.

Class 2: Feedback/Input did not adequately indicate ____

© Copyright 2024 John Thomas
STPA Step 4: Class 2 Scenario Archetype

**UCA-2:**
*<Controller>* provides *<Control Action> when* *<Context>*

**Discussion:** The “Why” answers may come from SMEs, not the STPA practitioner. You may not be an expert in the system. The point is for you to use this framework to ask questions and approach the SMEs to find these answers.

**Class 2:**
Feedback/Input did not adequately indicate ____

**Why? (non-failures)**
- Because the ____ metric is estimated from ____ that uses different ____
- Because the ____ feedback is outdated due to ____ second feedback delay
- Etc.

© Copyright 2024 John Thomas
STPA Step 4: Class 2 Scenario Archetype

UCA-2: 
<Controller> provides <Control Action> when <Context> 

**Discussion:** This is the generic form for the Class 2 Scenario Archetype.

Class 2 Scenario Archetype

**Inadequate Feedback**

**Output:**
<Feedback/Input> to <Controller> did not adequately indicate <Context> 

**Input:** <Process> is actually <Context> 

Controller

Sensor

- Inadequate operation
- Inadequate or missing feedback
- Feedback delays
- Incorrect or no information provided
- Measurement inaccuracies
- Feedback delays

Controlled Process

- Component failures
- Changes over time

Unidentified or out-of-range disturbance

Process output contributes to system hazard
Scenario Archetypes

**Class 1 Scenario Archetype:**
Inadequate Controller Behavior
- `<Controller>` provides _____ when _____
- `<Input>` to `<Controller>` correctly indicated _____

**Class 2 Scenario Archetype:**
Inadequate feedback/information
- `<Feedback/Input>` to `<Controller>` did not adequately indicate `<Context>`
- `<Process>` is actually `<Context>`
Example: Mars Polar Lander

Thruster Controller

Control Algorithm
Process Model (beliefs)

Disable Thruster
Touchdown

Lander
Mars Polar Lander

**Unsafe Control Action (UCA):**
Computer provides Disable-Thruster cmd when spacecraft is in the air

**Process Model:**
Incorrectly believes spacecraft is on ground

**Feedback:**
Touchdown indication received when in air

**Physical Interaction:**
Simultaneous leg vibration during leg deployment

**Thruster Controller**
- Control Algorithm
- Process Model (beliefs)

**Lander**
- Disable Thruster
- Touchdown
UCA-2: Thruster Controller provides Disable-Thruster Cmd when spacecraft is in the air

Example: Mars Polar Lander

Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it’s in the air

Class 2 Scenario Archetype: Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>
UCA-2: Thruster Controller provides Disable-Thruster Cmd when spacecraft is in the air.

Example: Mars Polar Lander

Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it’s in the air

Class 2 Scenario Archetype: Inadequate feedback/information
- Touchdown feedback does not indicate it’s in air
- <Process> is actually <Context>
UCA-2: Thruster Controller provides Disable-Thruster Cmd when spacecraft is in the air.

Example: Mars Polar Lander

Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it’s in the air

Class 2 Scenario Archetype: Inadequate feedback/information
- Touchdown feedback does not indicate it’s in air
- Lander is actually in the air
Scenario Archetypes

Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype:
Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

Ask: What can cause this Scenario Archetype?

Consider:
1) Failure causes
2) No failures
Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it’s in the air

Class 2 Scenario Archetype: Inadequate feedback/information
- Touchdown feedback does not indicate it’s in air
- Lander is in the air

What does “no failure” mean?
- It means the sensors, etc., worked as specified

What does “touchdown sensor worked” mean?
- It means the sensor output was as specified for the sensor input

We know the sensor feedback was [NOT IN AIR]. What was the input?
- If output = [NOT IN AIR], then input = [VIBRATION > X]
**Class 1 Scenario Archetype:**
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

**Class 2 Scenario Archetype:**
Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

**Class 3 Scenario Archetype:**

**Class 4 Scenario Archetype:**

**Why?**
- Because the ___ metric is estimated from ___ that uses different ___
- Because the ___ feedback is outdated due to ___ second feedback delay
- Etc.

© Copyright 2024 John Thomas
Refining Class 2 Scenario Archetype: Inadequate Feedback/Information

**Common causes of Scenario Archetype 2:**

- Feedback/info missing from design/concept
- Feedback/info not provided
- Conflicting feedback/info
- Incorrect feedback/info provided
- Too early or too late (delayed) feedback/info
- Measurement inaccuracies
- Dropouts
- Corruption
- Content incomplete
- Feedback/info provided in a way the controller can’t use
- Overloaded or too much feedback/info
- Etc.
Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype: Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Too early or too late info
<Feedback> can be sent too early before ____ has occurred.
UCA-2: 
<Controller> provides <Control Action> when <Context> 

Scenario Archetypes

Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

Class 2 Scenario Archetype: Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Refined Scenarios

No info provided
<Feedback> is not provided when ____ is initialized, reset, or on power up.
Scenario Archetypes

Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype: Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype:

Class 4 Scenario Archetype:

Discussion: These scenarios are used to help us think about mitigations. E.g., <Controller> currently has no feedback to indicate process overloading, so the design doesn’t currently have any way to correct it.
Four Classes of Formal Scenarios

Class 1
Controller -> Feedback or other inputs -> Controlled Process

Class 2
Controller -> Feedback or other inputs -> Controlled Process

Class 3
Controller -> Feedback or other inputs -> Controlled Process

Class 4
Controller -> Feedback or other inputs -> Controlled Process
Class 3 Scenario Archetype: Inadequate Control Execution

We need to look at how the UCA can be emulated due to interactions in this region.

Constructing Scenario Archetype 3:
- Suppose the UCA did not happen (invert the UCA). The controller provided a “safe” control action.
- BUT... something happened on the control path making it as if the UCA had occurred.
STPA Step 4: Class 3 Scenario Archetype

UCA-2: <Controller> provides <Control Action> when <Context>

Class 3 Scenario Archetype: Inadequate Control Execution
- <Controller> does not provide ____
- <Process> receives ____

This behavior would emulate the UCA, even if our controller does the right thing.

How could this happen? Other controllers? Spoofing? Other causes?
Scenario Archetypes

Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides ______ when ______
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype: Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype: Inadequate Control Execution
- <Controller> does not provide <Control Action> when <Context>
- <Process> receives a <Control Action> when <Context>

© Copyright 2024 John Thomas
Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it’s in the air

Class 2 Scenario Archetype: Inadequate feedback/information
- Touchdown feedback does not indicate it's in air
- Lander is in the air

Class 3 Scenario Archetype: Inadequate Control Execution
- <Controller> does not provide <Control Action> when <Context>
- <Process> receives a <Control Action> when <Context>

Example: Mars Polar Lander

UCA-2: Thruster Controller provides Disable-Thruster Cmd when spacecraft is in the air
Example: Mars Polar Lander

Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it's in the air

Class 2 Scenario Archetype: Inadequate feedback/information
- Touchdown feedback does not indicate it's in air
- Lander is in the air

Class 3 Scenario Archetype: Inadequate Control Execution
- Controller does not provide Disable-Thruster Cmd when spacecraft is in the air
- Lander receives Disable-Thruster Cmd when spacecraft is in the air
Scenario Archetypes

Class 1 Scenario Archetype: 
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype: 
Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype: 
Inadequate Control Execution
- <Controller> does not provide <Control Action> when <Context>
- <Process> receives a <Control Action> when <Context>

Class 4 Scenario Archetype:

Refined Scenarios

Ask: What can cause this Scenario Archetype?
Scenario Archetypes

Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides ______ when ______
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype:
Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype:
Inadequate Control Execution
- <Controller> does not provide <Control Action> when <Context>
- <Process> receives a <Control Action> when <Context>

Class 4 Scenario Archetype:

Why?
- Some other <Controller> could generate <Control Action> and send it to ___
- <Controller> sends <Control Action> with Ignore bit set, but ______.
- Etc.

Discussion: How would <Controlled Process> know to ignore this? It might not know.
Four Classes of Formal Scenarios

Class 1
- Controller
- Feedback or other inputs
- GOOD
- UNSAFE
- Control Actions
- Controlled Process

Class 2
- Controller
- Feedback or other inputs
- UNSAFE
- Control Actions
- Controlled Process

Class 3
- Controller
- Feedback or other inputs
- UNSAFE
- GOOD
- Control Actions
- Controlled Process

Class 4
- Controller
- Feedback or other inputs
- Control Actions
- Controlled Process
- GOOD
- UNSAFE
UCA-2: <Controller> provides <Control Action> when <Context>

Class 4 Scenario Archetype: Inadequate Process Behavior

We need to look at how the UCA can be emulated due to interactions in this region.

Constructing Scenario Archetype 4:
- Suppose the UCA did not happen (invert the UCA), and was not received by the controlled process.
- BUT... something happened with the controlled process and its other interactions making it as if the UCA had been provided.
STPA Step 4: Class 4 Scenario Archetype

UCA-2: <Controller> provides <Control Action> when <Context>

Controller

Actuator
Inadequate operation

Sensor
Inadequate operation

Controller

Class 4 Scenario Archetype: Inadequate Process Behavior
- <Process> does not receive ____
- <Process> does ____ anyway

Controller

Component failures
Changes over time

Process input missing or wrong

Unidentified or out-of-range disturbance

Process output contributes to system hazard
Scenario Archetypes

Class 1 Scenario Archetype: Inadequate Controller Behavior
- <Controller> provides ______ when ______
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype: Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype: Inadequate Control Execution
- <Controller> does not provide <Control Action> when <Context>
- <Process> receives a <Control Action> when <Context>

Class 4 Scenario Archetype: Inadequate process behavior
- <Process> does not receive a <Control Action> when <Context>
- <Process> applies <Control Action> when <Context>
Example: Mars Polar Lander

Class 1 Scenario Archetype: Inadequate Controller Behavior
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it’s in the air

Class 3 Scenario Archetype: Inadequate Control Execution
- Controller does not provide Disable-Thruster Cmd when spacecraft is in the air
- Lander receives Disable-Thruster Cmd when spacecraft is in the air

Class 4 Scenario Archetype: Inadequate process behavior
- <Process> does not receive a <Control Action> when <Context>
- <Process> applies <Control Action> when <Context>

© Copyright 2024 John Thomas
Example: Mars Polar Lander

**Class 1 Scenario Archetype: Inadequate Controller Behavior**
- Controller provides Disable-Thruster Cmd when spacecraft is in the air
- Touchdown Input to Controller correctly indicated it’s in the air

**Class 3 Scenario Archetype: Inadequate Control Execution**
- Controller does not provide Disable-Thruster Cmd when spacecraft is in the air
- Lander receives Disable-Thruster Cmd when spacecraft is in the air

**Class 4 Scenario Archetype: Inadequate process behavior**
- Thrusters do not receive Disable command when spacecraft is in the air
- Thrusters are disabled (when in the air)

© Copyright 2024 John Thomas
Scenario Archetypes

**Class 1 Scenario Archetype:**
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated _____

**Class 2 Scenario Archetype:**
Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

**Class 3 Scenario Archetype:**
Inadequate Control Execution
- <Controller> does not provide <Control Action> when <Context>
- <Process> receives a <Control Action> when <Context>

**Class 4 Scenario Archetype:**
Inadequate process behavior
- <Process> does not receive a <Control Action> when <Context>
- <Process> applies <Control Action> when <Context>

Ask: What can cause this Scenario Archetype?
Class 1 Scenario Archetype:
Inadequate Controller Behavior
- <Controller> provides _____ when _____
- <Input> to <Controller> correctly indicated ______

Class 2 Scenario Archetype:
Inadequate feedback/information
- <Feedback/Input> to <Controller> did not adequately indicate <Context>
- <Process> is actually <Context>

Class 3 Scenario Archetype:
Inadequate Control Execution
- <Controller> does not provide <Control Action> when <Context>
- <Process> receives a <Control Action> when <Context>

Class 4 Scenario Archetype:
Inadequate process behavior
- <Process> does not receive a <Control Action> when <Context>
- <Process> applies <Control Action> when <Context>

Why?
• <Process> may mechanically ___
• <Process> may run out of ___
• <Process> may see that ___ is full, in which case <Process> will automatically ____; which results in ____.
• If <Process> is in ___ mode, then all ___ will be ignored.
## Scenario Archetype Generation

### Scenario Archetype Table:

<table>
<thead>
<tr>
<th>Scenario Class 1: Unsafe Controller Behavior</th>
<th>Control Action</th>
<th>Scenario Archetype Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCA type 1: not providing causes hazard (UCA-#)</td>
<td>Not providing causes hazard</td>
<td>(Thomas, 2016), (Thomas, 2017), (Cabosky, 2020)</td>
</tr>
<tr>
<td>Scenario Class 2: Unsafe Feedback Path</td>
<td>Providing causes hazard</td>
<td>© Copyright 2024 John Thomas</td>
</tr>
<tr>
<td>Scenario Class 3: Unsafe Control Path</td>
<td>Too early, too late, out of order causes hazard (UCA-#)</td>
<td></td>
</tr>
<tr>
<td>Scenario Class 4: Unsafe Controlled Process Behavior</td>
<td>Stopped Too Soon / Applied too long causes hazard (UCA-#)</td>
<td></td>
</tr>
</tbody>
</table>
STPA Scenarios should cover:
How to run an STPA project

Let’s discuss who would do this and how they would coordinate with others
STPA Project Participants

**STPA Core Team**

- **STPA Practitioners**
  - Perform majority of STPA work
  - Interdisciplinary team
  - Should have STPA training and certification, but may not be experts

- **STPA Facilitator**
  - The STPA expert
  - Provide STPA method guidance (and other responsibilities)

- **Other SMEs**
  - Provide specialized domain knowledge as needed by team
  - May have little or no STPA familiarity
  - May not be actively involved in STPA, but must be accessible by team