Safety and Security Analysis Using STPA

Jingyue Li
Associate Professor
Department of Computer Science
Norwegian University of Science and Technology
Outline

- Why STAMP and STPA?
- STPA for safety analysis
- STPA for security analysis
- STPA – agile and cost effectiveness
- Summary
Most safety analysis tools are all 40-60 years old. Our technology is very different today.

- Introduction of computer control
- Exponential increase in interaction and complexity
- Lots of new technology

Mechanical and electrical components dominate.

More instrumentation and software control.
Traditional accident causation model: accidents as chains of failure events

Heinrich Domino Theory (1930) (Teori Domino)
The “accidents as chains of failure events” model

• Safety analysis
  – FTA, FMEA/FMECA, Event Tree, HZAOP, etc.

• Safety design (concentrates on component failure)
  – High component integrity
  – Redundancy and overdesign
  – Barriers (to prevent failure propagation)
  – Fail-safe design
  – Operational procedures
  – …
Traditional approaches do not handle well component interaction accidents

- Component interaction accidents
  - No component stops working
  - Design is wrong
  - Components (and humans) do not fit together
  - Especially for indirect and non-linear interactions
  - Social-technical aspects
Multiple controller problem

- Conflicting control actions
- Overriding between commands
  - An unsafe command overrides a safe one
- “Someone else has done (will do)”
  - Each controller thinks the other has done (will do) and nobody does
- Etc…
An example of wrong interaction

- One pilot executed a planned **test** by aiming at aircraft in front and firing a **dummy** missile.

- **Nobody** involved knew that the **software was designed** to substitute a different missile if the one that was commanded to be fired was not in a good position.

- In this case, there was an antenna between the dummy missile and the target so the **software** decided to fire a **live** missile located in a different (better) position instead.

- **Accident**: a live missile was fired instead of the dummy missile!
STAMP (Systems-Theoretic Accident Model and Processes): A new accident causation model

• **STAMP expands** the traditional accident causation model
  – Accidents are more than a chain of directly related failure events
  – Accidents involve more complex dynamic processes
  – Safety can only be treated adequately in their entirety (all social and technical aspects)

• Treat accident as a **control** problem, **not just a failure** problem

  “Prevent failure”

  “Enforce safety constraints (e.g. Two aircrafts must not violate minimum separation)”
STAMP is a new accident causality model

Applications
- System engineering (e.g. Specification, Safety-Guided Design, Design principles)
- Risk management
- Operations
- Management Principles or Organizational design
- Regulations

Methods
- Hazard analysis
  - STPA
  - (System Theoretic Process Analysis)
- Security Analysis
  - STPA-Sec
- Accident/Event Analysis
  - CAST
  - (Causal Analysis using System Theory)
- Early Concept Analysis
  - STECA

STAMP: Theoretical Causality Model
Outline

• Why STAMP and STPA?
  ▶ STPA for safety analysis
• STPA for security analysis
• STPA – agile and cost effectiveness
• Summary
STPA has three key concepts

- Safety constraint
- The hierarchical control structure
- Process models
STPA applied in train door control system – operation control structure

Control Command: Open door

Close door

Driver

Software Controller

Mental process model

Control software process model

Process model variables

Process model variable states

Door position

Fully open/Fully closed/partially open

Train condition

Normal/Emergency

Train motion

Stopped/Moving

Door position

Train position

Aligned with plat. / not aligned with plat.

Train motion

Stopped/Moving

Door position sensor

Train speed sensor

Train position sensor

Door actuator

Process model variables
STPA steps

1. Identify Accident & hazards & safety constraint
2. Draw the control structure
3. Identify Unsafe Control Actions (UCA)
4. Identify casual factors of UCA

Unsafe Control Actions: Controller’s final commands to actuators that violate safety constraints.

STPA systematically reveals the unsafe control actions (UCA) and the causal factors.
STPA applied in train door control system – operation control structure

Control Command: Open door
Close door

Control Action (CA)
• CA1: Open door
• CA2: Stop opening door
• CA3: Close door
• CA4: Stop closing door

Driver
Mental process model
- Process model variables
- Train condition
- Normal/Emergency
- Train motion
- Stopped/Moving
- Door position
- Fully open/Fully closed/partially open

Software Controller
Control software process model
- Process model variables
- Door position
- Fully open/Fully closed/partially open
- Train position
- Aligned with plat./not aligned with plat.
- Train motion
- Stopped/Moving

Door actuator

Train motion
- Door position
- Speed
- Position

Door position sensor
Train speed sensor
Train position sensor

Process model variables
STPA applied in train door control system – how to identify UCA?

STPA evaluates each Control action for all combinations of Process Model Variable States.

Under each combination of process model variable state, STPA will evaluate if any of the following four scenarios will be safe or unsafe.

1) A control action required is not provided
2) A control action is provided
3) A control action is provided too late, too early, or out of sequence
4) A control action is stopped too soon or applied too long
STPA applied in train door control system – identify if a certain CA is hazardous

<table>
<thead>
<tr>
<th>Controller</th>
<th>Door control system</th>
<th>H1</th>
<th>Door opens when the train is in motion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control Action</strong></td>
<td><strong>Open door</strong></td>
<td>H2</td>
<td>Door opens while not aligned with station platform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H3</td>
<td>Door cannot be opened for emergency evacuation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H4</td>
<td>Door closes while someone is in the doorway</td>
</tr>
</tbody>
</table>

**Process Model Variables**

| | | | CA NOT provided | CA provided | CA provided too late/early | CA stopped too late/early |
| | | | Train motion | Emergency (Yes/No) | Train position (Aligned) | |
| 1 | Stopped | Yes | No | H3 | Too late (H3) | Too early (H3) |
| 2 | Stopped | Yes | Yes | H3 | Too late (H3) | Too early (H3) |
| 3 | Stopped | No | No | H3 | Too early/late (H2) | |
| 4 | Stopped | No | Yes | H2 | Too early (H2) | |
| 5 | Stopped | No | Yes | H1, H2 | Too early (H1, H2) | |
| 6 | Moving | Yes | No | H1 | Too early (H1) | |
| 7 | Moving | Yes | Yes | H1, H2 | Too early (H1, H2) | |
| 8 | Moving | No | Yes | H1, | Too early (H1) | |

How can this happen?
A classification of causal factors leading to hazards

1. Controller does not provide a CA correctly
   - Control Algorithm (flaws in creation, process changes, incorrect modification or adaptation)
   - Process Model (inconsistent, incomplete, or incorrect)
   - Communication with another controller (missing or wrong)
   - Received Feedback (inadequate, missing, or delayed)

2. Proper controls provided but NOT followed
   - Inadequate operation
   - Provided Feedback (incorrect, no information provided, measurement inaccuracies, delays)
   - Process output contributes to system hazard
   - Unidentified or out-of-range disturbance

"Guidewords"
Outline

• Why STAMP and STPA?
• STPA for safety analysis
  ➢ STPA for security analysis
• STPA – agile and cost effectiveness
• Summary
STPA-Sec

Attacks targeting at sensors, actuators, and control units

Interaction between components

Failure of single component

Classical hazard methods

STPA-Sec

STPA

Need to do threat modeling
## STPA + STPA-Sec

<table>
<thead>
<tr>
<th>Controller</th>
<th>Door control system</th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>H4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open door</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Control Action

- Door opens when the train is in motion
- Door opens while not aligned with station platform
- Door cannot be opened for emergency evacuation
- Door closes while someone is in the doorway

### Process Model Variables

<table>
<thead>
<tr>
<th>Train motion</th>
<th>Emergency (Yes/No)</th>
<th>Train position (Aligned)</th>
<th>CA NOT provided</th>
<th>CA provided</th>
<th>CA provided too late/early</th>
<th>CA stopped too late/early</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stopped</td>
<td>Yes</td>
<td>No</td>
<td>H3</td>
<td></td>
<td>Too late (H3)</td>
<td>Too early (H3)</td>
</tr>
<tr>
<td>Stopped</td>
<td>Yes</td>
<td>Yes</td>
<td>H3</td>
<td></td>
<td>Too late (H3)</td>
<td>Too early (H3)</td>
</tr>
<tr>
<td>Stopped</td>
<td>No</td>
<td>No</td>
<td>H2</td>
<td></td>
<td>Too early/late (H2)</td>
<td></td>
</tr>
<tr>
<td>Stopped</td>
<td>No</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Too early (H2)</td>
</tr>
<tr>
<td>Moving</td>
<td>Yes</td>
<td>No</td>
<td>H1, H2</td>
<td></td>
<td>Too early (H1, H2)</td>
<td></td>
</tr>
<tr>
<td>Moving</td>
<td>Yes</td>
<td>Yes</td>
<td>H1</td>
<td></td>
<td>Too early (H1)</td>
<td></td>
</tr>
<tr>
<td>Moving</td>
<td>No</td>
<td>No</td>
<td>H1, H2</td>
<td></td>
<td>Too early (H1, H2)</td>
<td></td>
</tr>
<tr>
<td>Moving</td>
<td>No</td>
<td>Yes</td>
<td>H1,</td>
<td></td>
<td>Too early (H1)</td>
<td></td>
</tr>
<tr>
<td>Moving but shows stopped</td>
<td>No</td>
<td>Yes</td>
<td>H1,</td>
<td>Too early (H1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moving</td>
<td>No</td>
<td>False aligned</td>
<td>H1, H2</td>
<td></td>
<td>Too early (H1, H2)</td>
<td></td>
</tr>
</tbody>
</table>

...
A classification of causal factors leading to hazards (with security)
Outline

• Why STAMP and STPA?
• STPA for safety analysis
• STPA for security analysis
  ➢ STPA – agile and cost effectiveness
• Summary
Agility

• Changes of process model variables
  – Add / remove / change control components
  – Add / remove / change interfaces

• Changes of threat models
Cost effectiveness

- State explosion
- Combinatorial testing methods

Number of variables involved in triggering software faults*

<table>
<thead>
<tr>
<th>Vars</th>
<th>Medical Devices</th>
<th>Browser</th>
<th>Server</th>
<th>NASA GSFC</th>
<th>Network Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>66</td>
<td>29</td>
<td>42</td>
<td>68</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>97</td>
<td>76</td>
<td>70</td>
<td>93</td>
<td>65</td>
</tr>
<tr>
<td>3</td>
<td>99</td>
<td>95</td>
<td>89</td>
<td>98</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>97</td>
<td>96</td>
<td>100</td>
<td>98</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>99</td>
<td>96</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*http://csrc.nist.gov/groups/SNS/acts/ftfi.htm
Summary

• STAMP and STPA has been applied in many domains

• STPA-Sec is developing

• Agility and cost-effectiveness will be key challenges