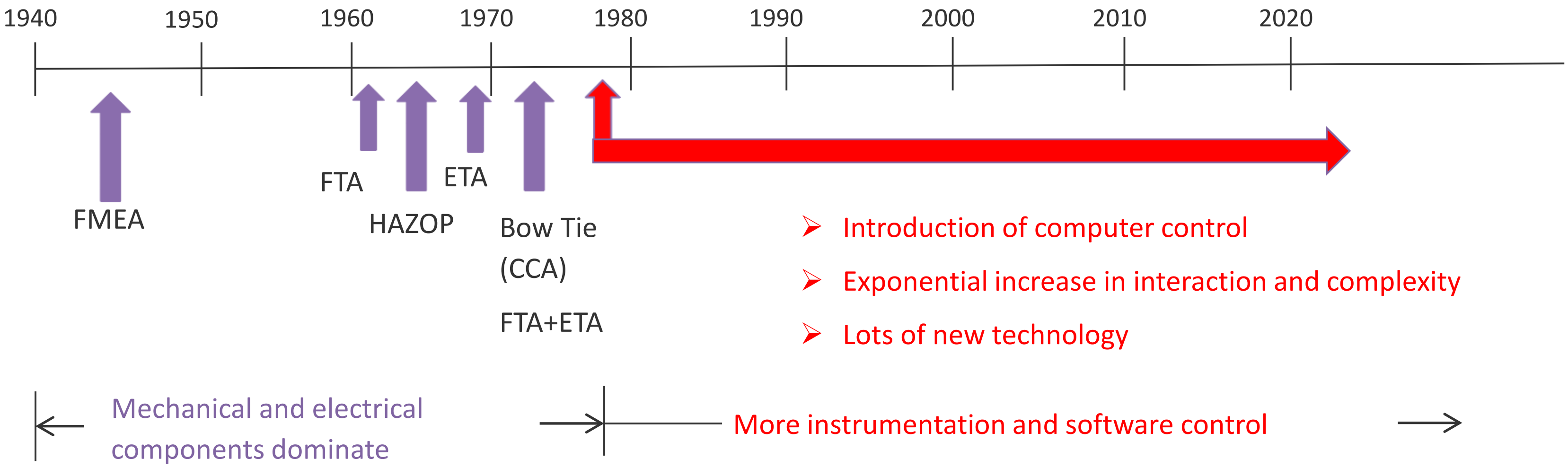
# Safety and Security Analysis Using STPA

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

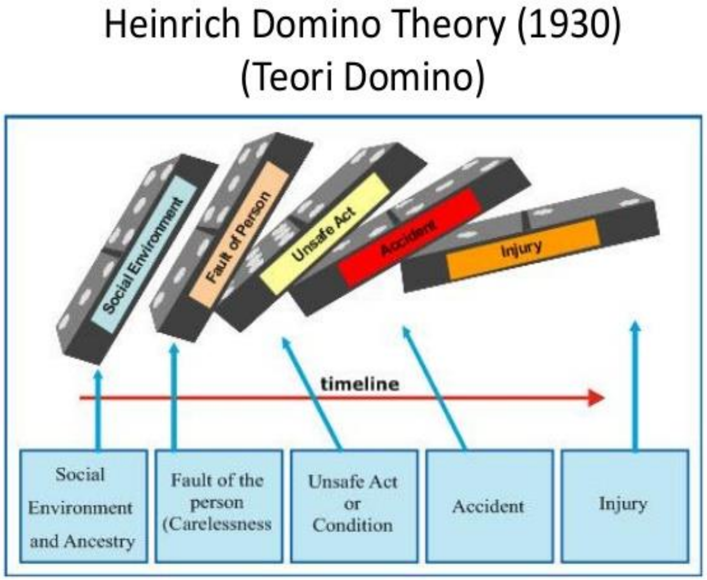


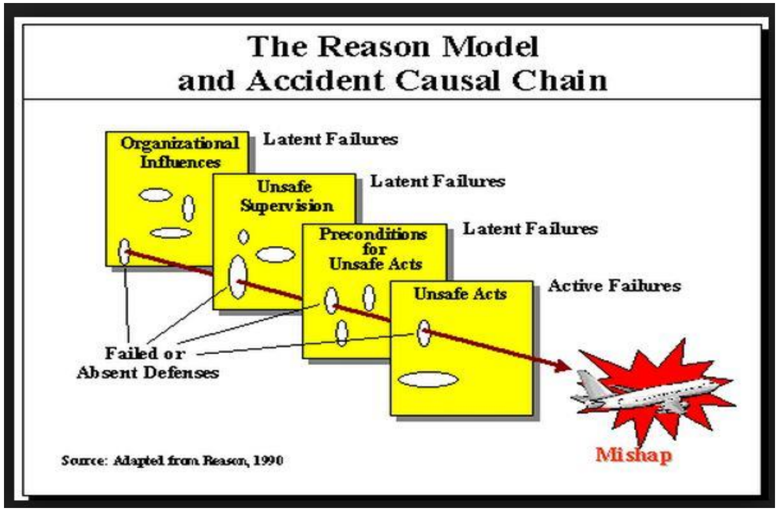
- ➢ Introduction of computer control
- ➢ Exponential increase in interaction and complexity
- ➢ Lots of new technology

Mechanical and electrical components dominate

More instrumentation and software control

# Traditional accident causation model: accidents as chains of failure events




The Reason Model
and Accident Causal Chain
Organizational Influences
Latent Failures
Unsafe Supervision
Latent Failures
Preconditions for Unsafe Acts
Latent Failures
Unsafe Acts
Active Failures
Failed or Absent Defenses
Mishap
Source: Adapted from Reason, 1990

# The “accidents as chains of failure events” model

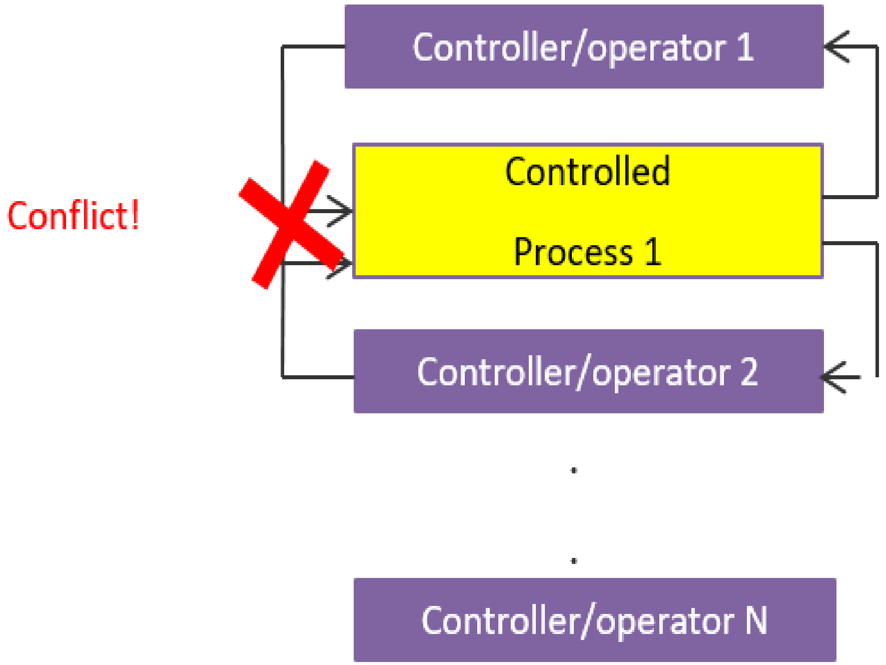
- Safety analysis
  - FTA, FMEA/FMECA, Event Tree, HZAOP, etc.

- 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)

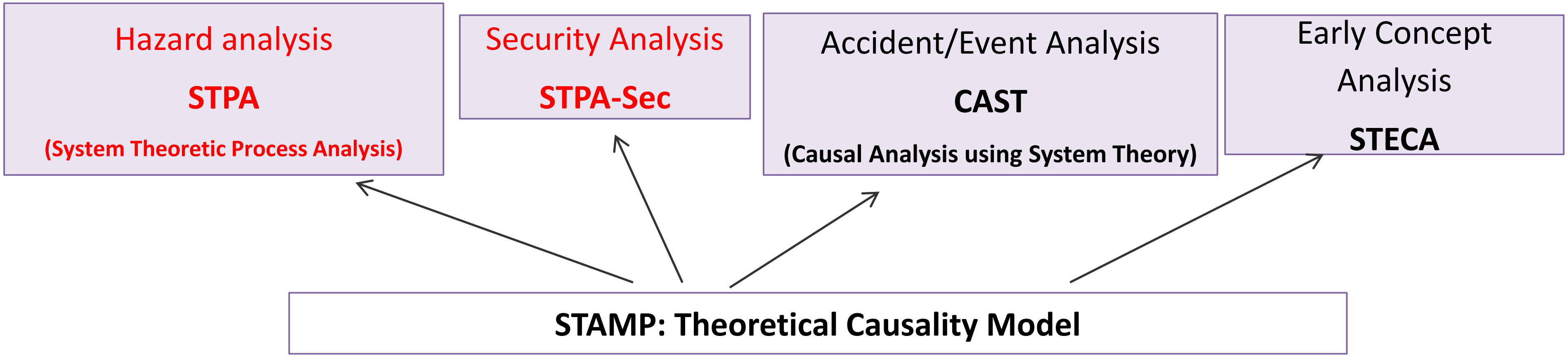
Risk management

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

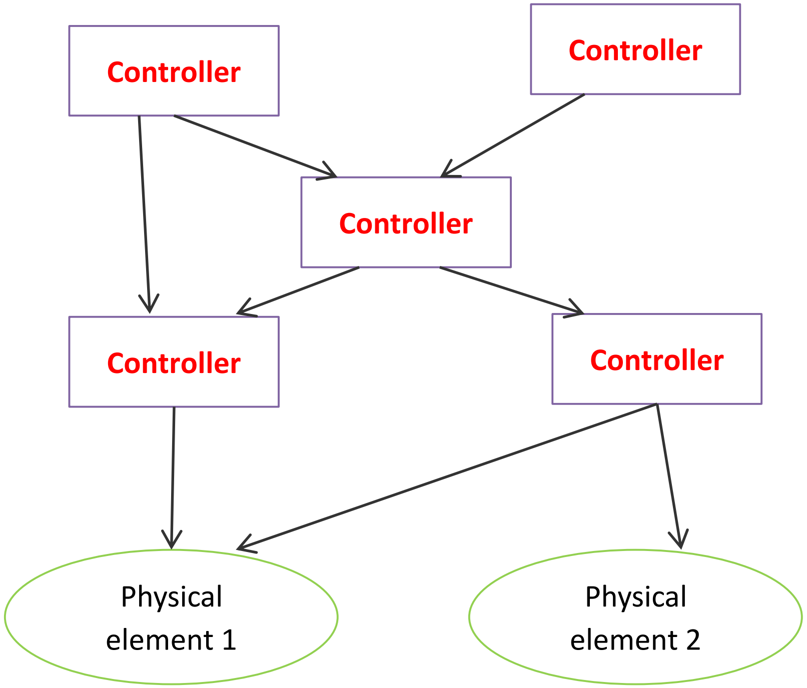
- Why STAMP and STPA?

➢ STPA for safety analysis

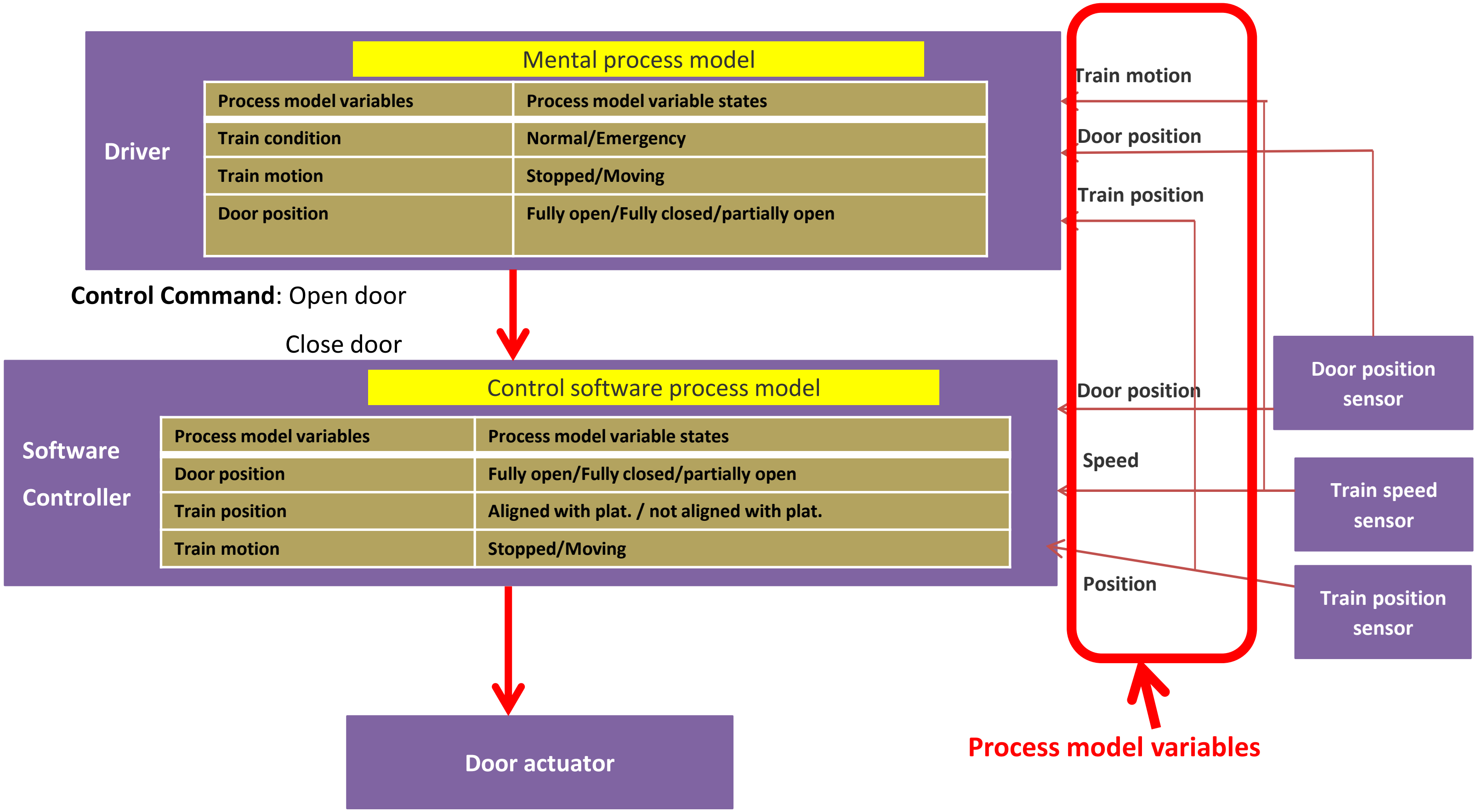
- STPA for security analysis
- STPA – agile and cost effectiveness
- Summary

# STPA has three key concepts

- Safety constraint
- The hierarchical control structure
- Process models



# STPA applied in train door control system – operation control structure



Driver

Mental process model

| Process model variables | Process model variable states |
|---|---|
| Train condition | Normal/Emergency |
| Train motion | Stopped/Moving |
| Door position | Fully open/Fully closed/partially open |

**Control Command**: Open door

Close door

Software Controller

Control software process model

| Process model variables | Process model variable states |
|---|---|
| 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

Train position

Door position

Speed

Position

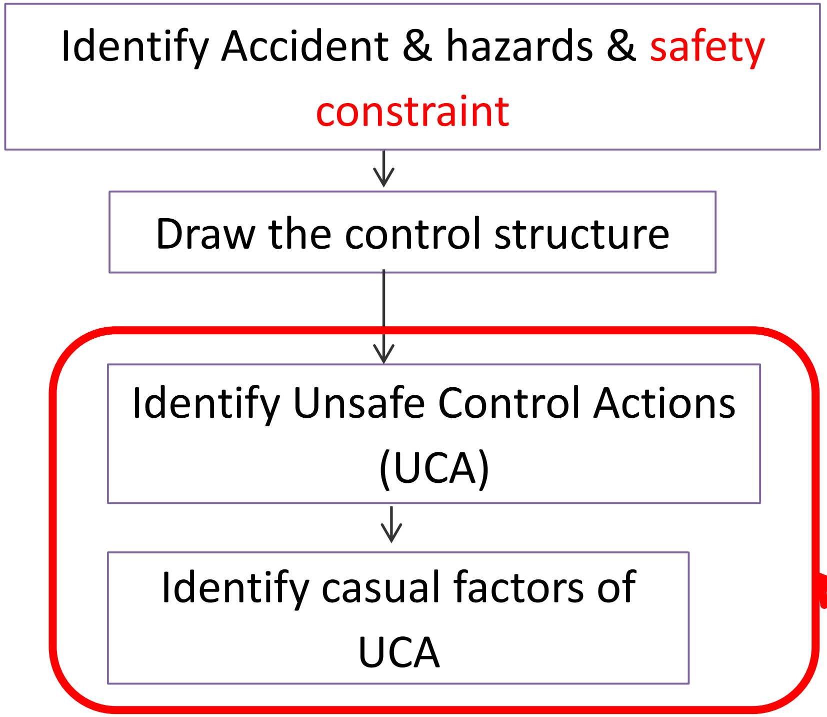
Door position sensor

Train speed sensor

Train position sensor

Process model variables

# STPA steps



Identify Accident & hazards & safety constraint

Draw the control structure

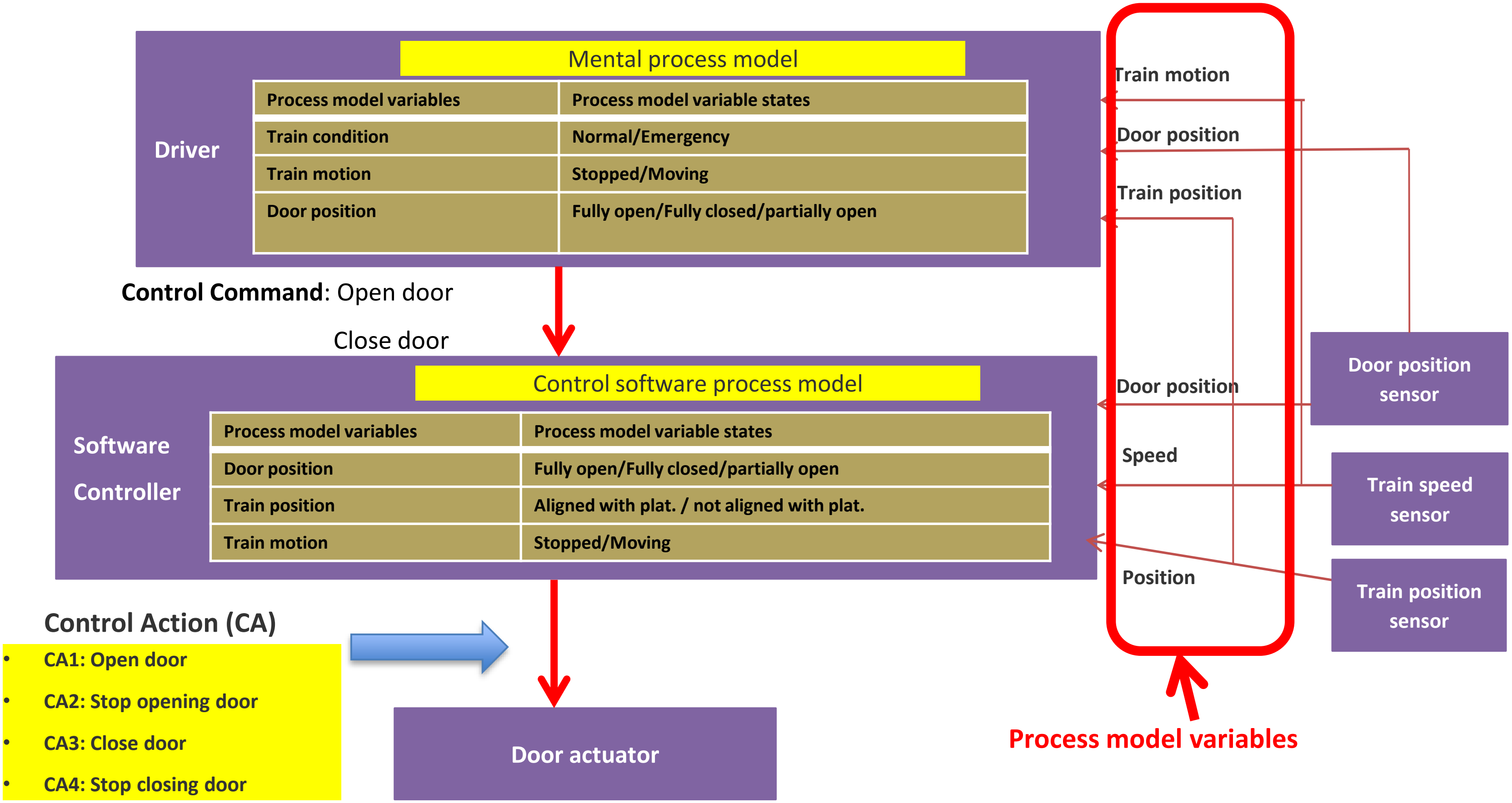
Identify Unsafe Control Actions (UCA)

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



| Process model variables | Process model variable states |
|---|---|
| Train condition | Normal/Emergency |
| Train motion | Stopped/Moving |
| Door position | Fully open/Fully closed/partially open |

**Control Command**: Open door

Close door

| Process model variables | Process model variable states |
|---|---|
| Door position | Fully open/Fully closed/partially open |
| Train position | Aligned with plat. / not aligned with plat. |
| Train motion | Stopped/Moving |

Control Action (CA)

- CA1: Open door
- CA2: Stop opening door
- CA3: Close door
- CA4: Stop closing door

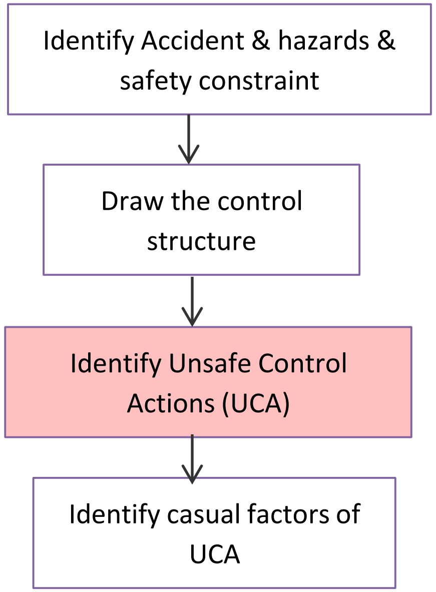
Process model variables

# 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 tool late, too early, or out of sequence
4) A control action is stopped too soon or applied too long


Identify Accident & hazards & safety constraint
Draw the control structure
Identify Unsafe Control Actions (UCA)
Identify casual factors of UCA

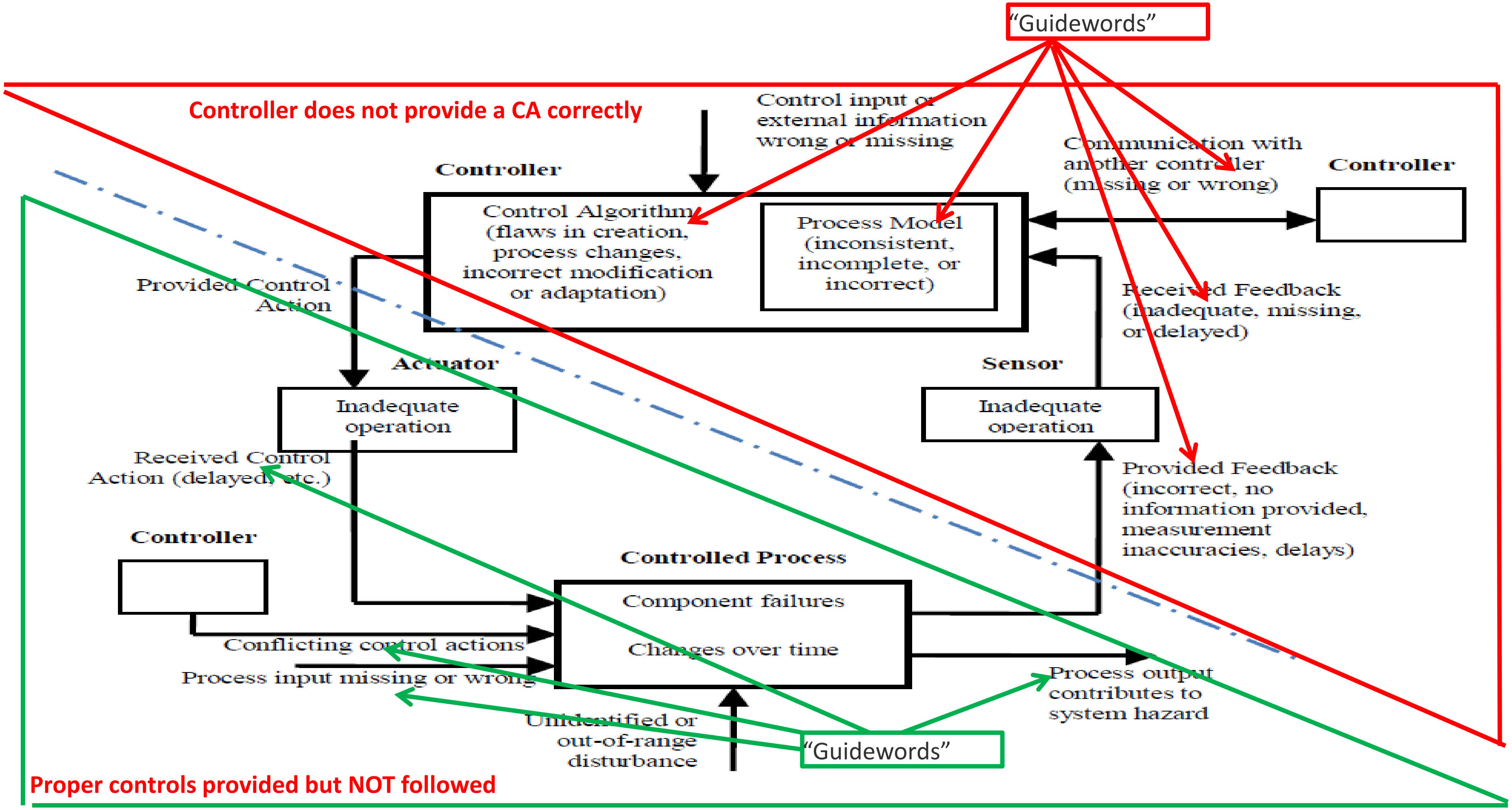
# STPA applied in train door control system – identify if a certain CA is hazardous

| | | | |
|---|---|---|---|
| Controller | Door control system | H1 | Door opens when the train is in motion |
| Control Action | Open door | H2 | Door opens while not aligned with station platform |
| | | H3 | Door cannot be opened for emergency evacuation |
| | | H4 | Door closes while someone is in the doorway |

| | Process Model Variables | | | Control Actions (CA) hazardous? | | | |
|---|---|---|---|---|---|---|---|
| | **Train motion** | **Emergency (Yes/No)** | **Train position (Aligned)** | **CA NOT provided** | **CA provided** | **CA provided too late/early** | **CA stopped too late/early** |
| 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 | | H2 | Too early/late (H2) | |
| 4 | Stopped | No | Yes | | | Too early (H2) | |
| 5 | Moving | Yes | No | | H1, H2 | Too early (H1, H2) | |
| 6 | Moving | Yes | Yes | | H1 | Too early (H1) | |
| 7 | Moving | No | No | | H1, H2 | Too early (H1, H2) | |
| 8 | Moving | No | Yes | | H1, | Too early (H1) | |

How can this happen?

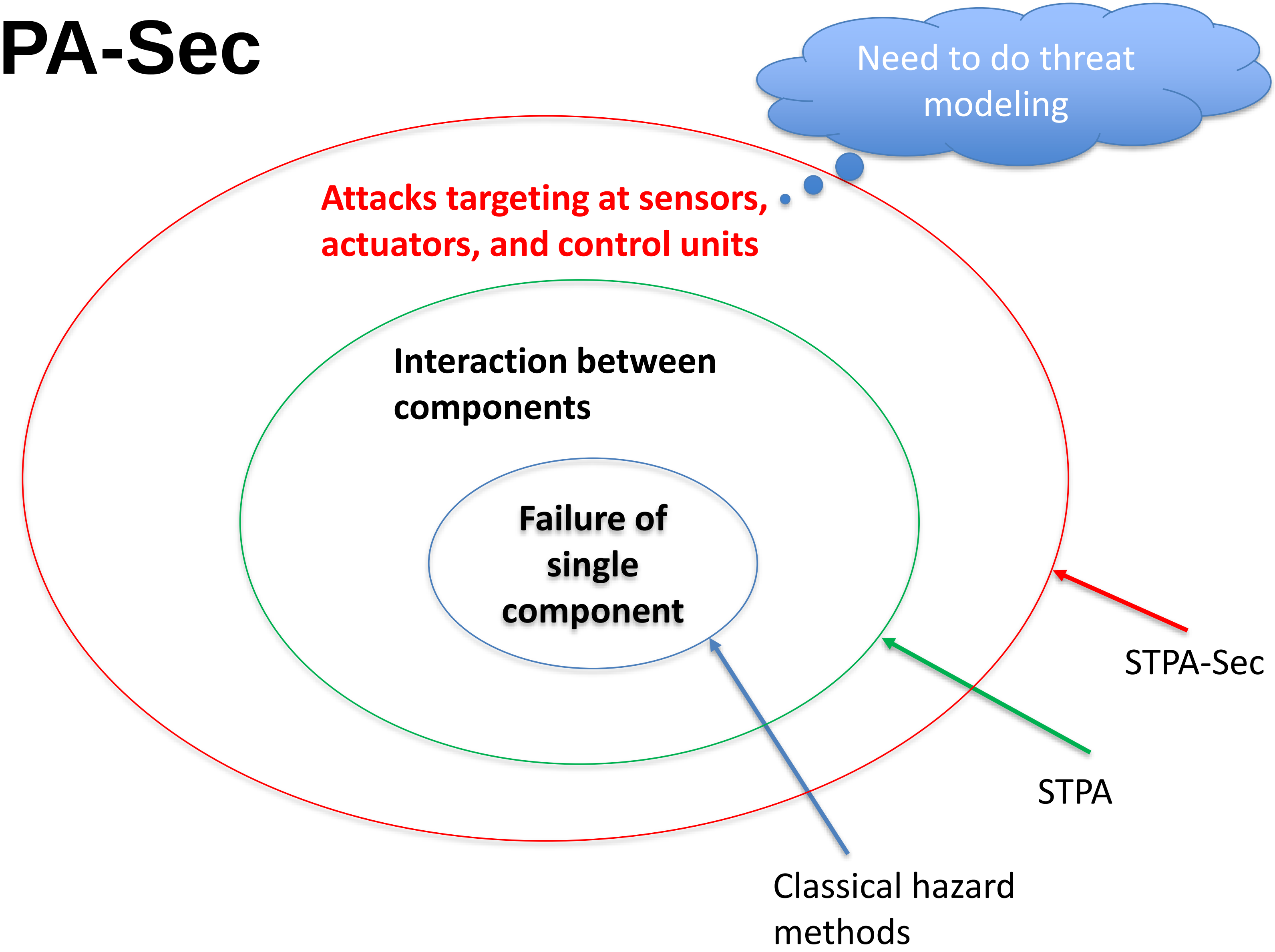
# A classification of causal factors leading to hazards


"Guidewords"
Controller does not provide a CA correctly
Control input or external information wrong or missing
Controller
Control Algorithm (flaws in creation, process changes, incorrect modification or adaptation)
Process Model (inconsistent, incomplete, or incorrect)
Communication with another controller (missing or wrong)
Controller
Provided Control Action
Actuator
Inadequate operation
Sensor
Inadequate operation
Received Feedback (inadequate, missing, or delayed)
Received Control Action (delayed, etc.)
Provided Feedback (incorrect, no information provided, measurement inaccuracies, delays)
Controller
Controlled Process
Component failures
Changes over time
Conflicting control actions
Process input missing or wrong
Process output contributes to system hazard
Unidentified or out-of-range disturbance
"Guidewords"
Proper controls provided but NOT followed

# Outline

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

# STPA-Sec



Need to do threat modeling

Attacks targeting at sensors, actuators, and control units

Interaction between components

Failure of single component

STPA-Sec

STPA

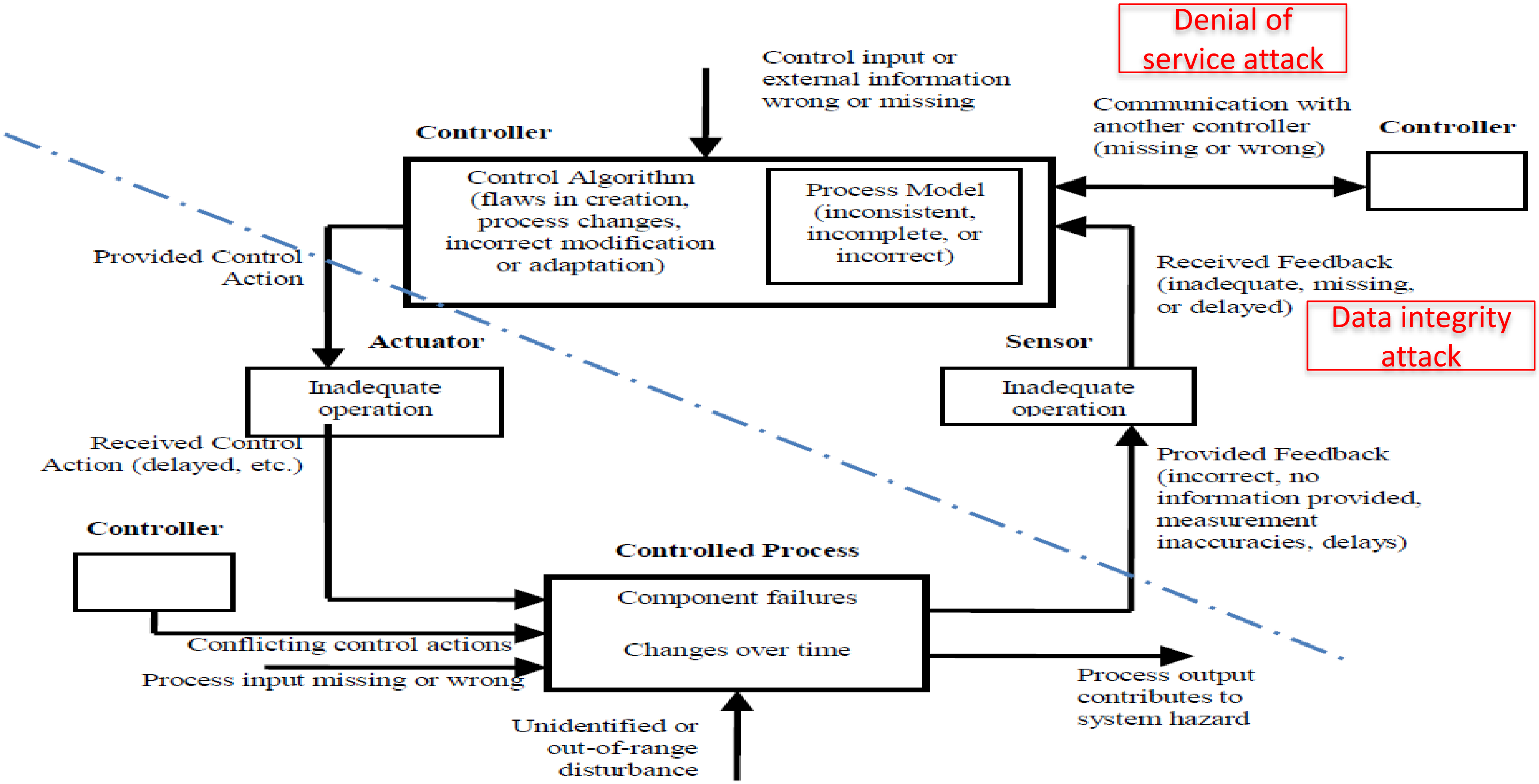
Classical hazard methods

# STPA + STPA-Sec

| | | | |
|---|---|---|---|
| Controller | Door control system | H1 | Door opens when the train is in motion |
| Control Action | Open door | H2 | Door opens while not aligned with station platform |
| | | H3 | Door cannot be opened for emergency evacuation |
| | | H4 | Door closes while someone is in the doorway |

| | Process Model Variables | | | Control Actions (CA) hazardous? | | | |
|---|---|---|---|---|---|---|---|
| | **Train motion** | **Emergency (Yes/No)** | **Train position (Aligned)** | **CA NOT provided** | **CA provided** | **CA provided too late/early** | **CA stopped too late/early** |
| 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 | | H2 | Too early/late (H2) | |
| 4 | Stopped | No | Yes | | | Too early (H2) | |
| 5 | Moving | Yes | No | | H1, H2 | Too early (H1, H2) | |
| 6 | Moving | Yes | Yes | | H1 | Too early (H1) | |
| 7 | Moving | No | No | | H1, H2 | Too early (H1, H2) | |
| 8 | Moving | No | Yes | | H1, | Too early (H1) | |
| 9 | Moving but shows stopped | No | Yes | | H1 | Too early (H1) | |
| 10 | Moving | No | False aligned | | H1, H2 | Too early (H1, H2) | |
| … | | | | | | | |

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


Denial of service attack
Control input or external information wrong or missing
Communication with another controller (missing or wrong)
Controller
Controller
Control Algorithm (flaws in creation, process changes, incorrect modification or adaptation)
Process Model (inconsistent, incomplete, or incorrect)
Provided Control Action
Received Feedback (inadequate, missing, or delayed)
Data integrity attack
Actuator
Inadequate operation
Sensor
Inadequate operation
Received Control Action (delayed, etc.)
Provided Feedback (incorrect, no information provided, measurement inaccuracies, delays)
Controller
Controlled Process
Component failures
Changes over time
Conflicting control actions
Process input missing or wrong
Process output contributes to system hazard
Unidentified or out-of-range disturbance

# Outline

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

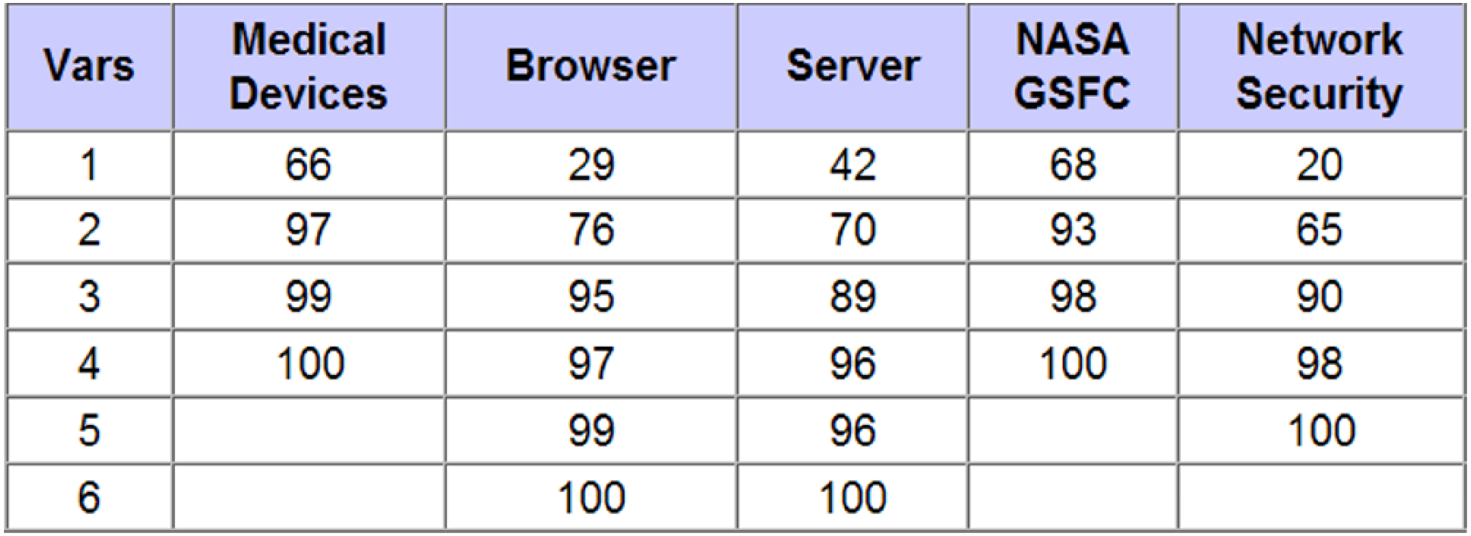
# Agility

- 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*



| Vars | Medical Devices | Browser | Server | NASA GSFC | Network Security |
|---|---|---|---|---|---|
| 1 | 66 | 29 | 42 | 68 | 20 |
| 2 | 97 | 76 | 70 | 93 | 65 |
| 3 | 99 | 95 | 89 | 98 | 90 |
| 4 | 100 | 97 | 96 | 100 | 98 |
| 5 |  | 99 | 96 |  | 100 |
| 6 |  | 100 | 100 |  |  |

*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