RESEARCH REVIEW 2025

Carnegie Mellon University Software Engineering Institute

Formalizing and Automating STPA with Robustness (FASR)

NOVEMBER 14–16, 2025

Sam Procter, Keaton Hanna, Lutz Wrage
Eunsuk Kang, Ian Dardik, Yining She

ADVANCING SOFTWARE for NATIONAL SECURITY
40 years
SOFTWARE ENGINEERING INSTITUTE

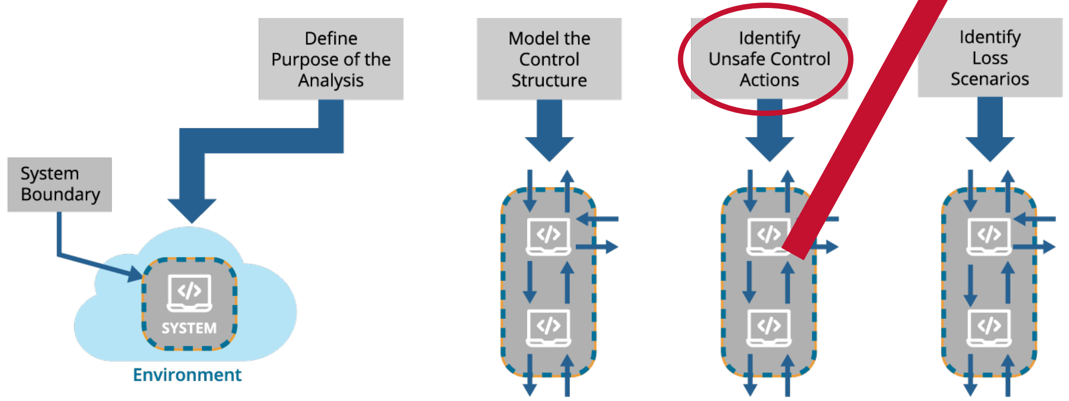
Innovation in Action

Document Markings

Background: Safety Analysis Is Slow, Hard, and You Don’t Know When You’re Done

Most critical systems are evaluated for safety before use, often using *hazard analysis* techniques.

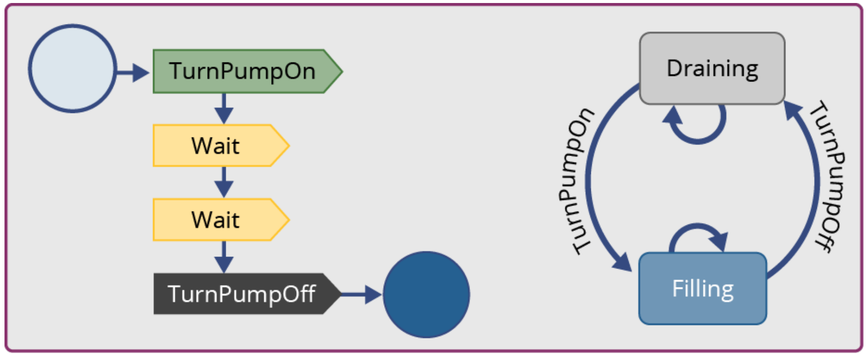
Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Too Early, Too Late, Out of Order	Stopped Too Soon, Applied Too Long
Brake	UCA-1: Brake Does Not Engage	UCA-2: Brake Engages During Takeoff	UCA-3: Brake Engages Too Late After Touchdown	UCA-4: Brake Disengages Before Safe Taxi Speed is Attained


Define Purpose of the Analysis
Model the Control Structure
Identify Unsafe Control Actions
Identify Loss Scenarios
System Boundary
SYSTEM
Environment

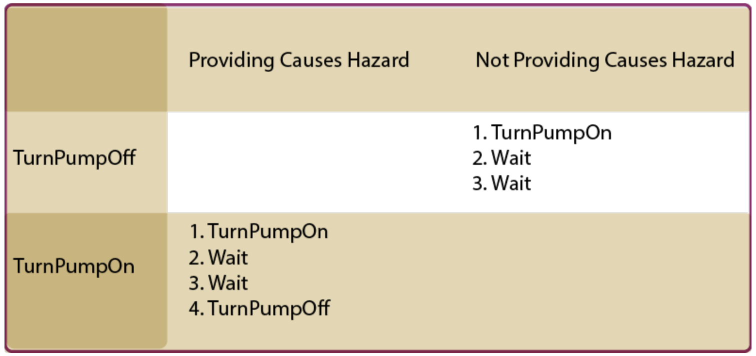
Research question: Can we use *formal methods* to improve analysis **speed** and **accuracy** while providing a **measure of completeness**?

Goal: Automatically Calculate Unsafe Control Actions from System Models

SysML

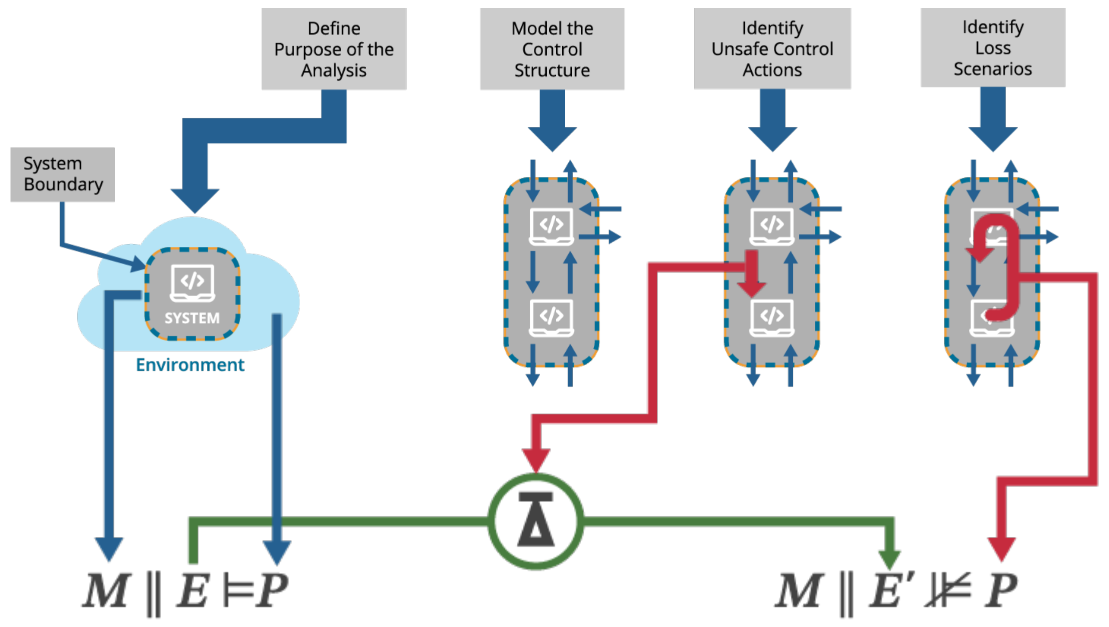


RAAML/STPA



	Providing Causes Hazard	Not Providing Causes Hazard
TurnPumpOff		1. TurnPumpOn 2. Wait 3. Wait
TurnPumpOn	1. TurnPumpOn 2. Wait 3. Wait 4. TurnPumpOff	

Robustness: Safety in the Presence of Environmental Deviations

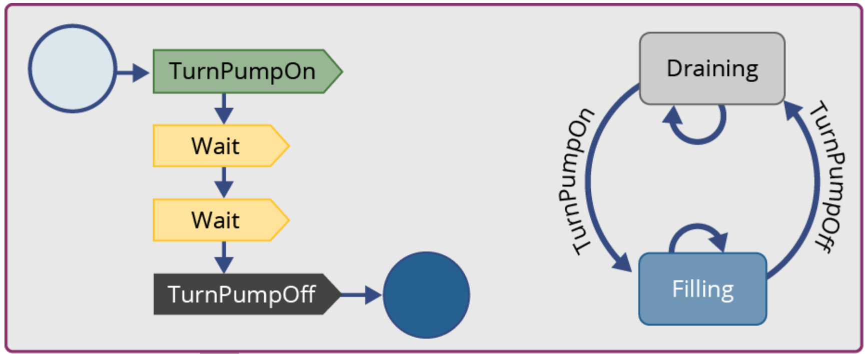

Define Purpose of the Analysis
Model the Control Structure
Identify Unsafe Control Actions
Identify Loss Scenarios
System Boundary
SYSTEM
Environment
$M \parallel E \models P$
$M \parallel E' \not\models P$

"… a system is *robust* with respect to a *property* and a particular set of *environmental deviations* if the system continues to satisfy the property even if the environment exhibits those deviations."

Zhang, Changjian, David Garlan, and Eunsuk Kang. 2020. "A Behavioral Notion of Robustness for Software Systems." *ESEC/FSE 2020 - Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (New York, NY, USA), November 8, 111–22. https://doi.org/10.1145/3368089.3409753.

Method: Translate SysML into Precise Specification

SysML



RAAML/STPA

	Providing Causes Hazard	Not Providing Causes Hazard
TurnPumpOff		1. TurnPumpOn 2. Wait 3. Wait
TurnPumpOn	1. TurnPumpOn 2. Wait 3. Wait 4. TurnPumpOff	

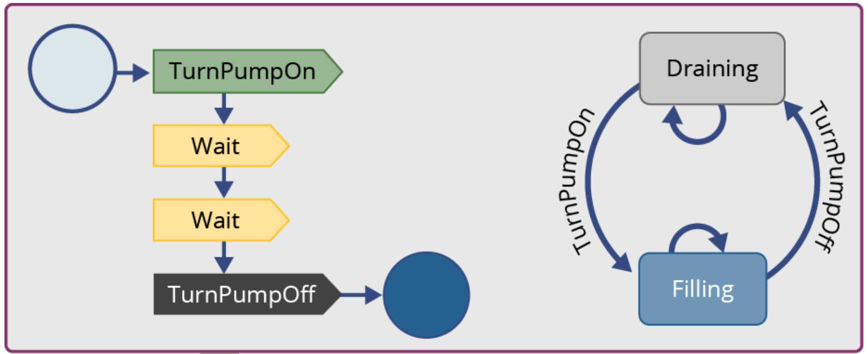
TLA+

```
Module Tank

Extends Integers

Variables WaterLevel, PumpOn
vars = <<waterLevel,
pumpOn>>

Init ==
   ∧ waterLevel = 0
   ∧ pumpOn = False
TurnPumpOn ==...
```

Method: Use CMU *Fortis* to Find Unsafe Behaviors

Carnegie Mellon University Software Engineering Institute

SysML



RAAML/STPA

	Providing Causes Hazard	Not Providing Causes Hazard
TurnPumpOff		1. TurnPumpOn 2. Wait 3. Wait
TurnPumpOn	1. TurnPumpOn 2. Wait 3. Wait 4. TurnPumpOff	

TLA+

```
Module Tank

Extends Integers

Variables WaterLevel, PumpOn
vars = <<waterLevel,
pumpOn>>

Init ==
   ∧ waterLevel = 0
   ∧ pumpOn = False
TurnPumpOn ==...
```

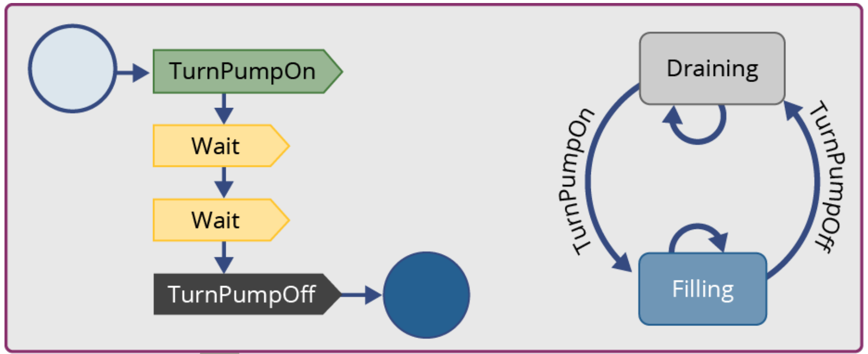
JSON

```
[
  {
    "goodTrace":   ["TurnPumpOn", "wait", "wait", "TurnPumpOff"]
    "badTrace":    ["TurnPumpOn", "wait", "wait", "wait"]
    "vioComp":     ["WaterTank"]
    "violnv":      ["NoOverflow"]
```

```
    "good Trace":  ["TurnPumpOn", "wait", "wait", "TurnPumpOff"]
    "badTrace":    ["TurnPumpOn", "wait", "wait", "TurnPumpOff", "TurnPumpOn", "wait"]
    "vioComp":     ["WaterTank"]
    "violnv":      ["NoOverflow"]
  }
]
```

Method: Categorize Behaviors with STPA Guidewords

SysML



RAAML/STPA

	Providing Causes Hazard	Not Providing Causes Hazard
TurnPumpOff		1. TurnPumpOn 2. Wait 3. Wait
TurnPumpOn	1. TurnPumpOn 2. Wait 3. Wait 4. TurnPumpOff	

TLA+

```
Module Tank

Extends Integers

Variables WaterLevel, PumpOn
vars = <<waterLevel,
pumpOn>>

Init ==
   ∧ waterLevel = 0
   ∧ pumpOn = False
TurnPumpOn ==...
```

JSON

```
[
  {
    "goodTrace":   ["TurnPumpOn", "wait", "wait", "TurnPumpOff"]
    "badTrace":    ["TurnPumpOn", "wait", "wait", "wait"]
    "vioComp":     ["WaterTank"]
    "violnv":      ["NoOverflow"]
```

```
    "good Trace":   ["TurnPumpOn", "wait", "wait", "TurnPumpOff"]
    "badTrace":     ["TurnPumpOn", "wait", "wait", "TurnPumpOff", "TurnPumpOn", "wait"]
    "vioComp":      ["WaterTank"]
    "violnv":       ["NoOverflow"]
  }
]
```

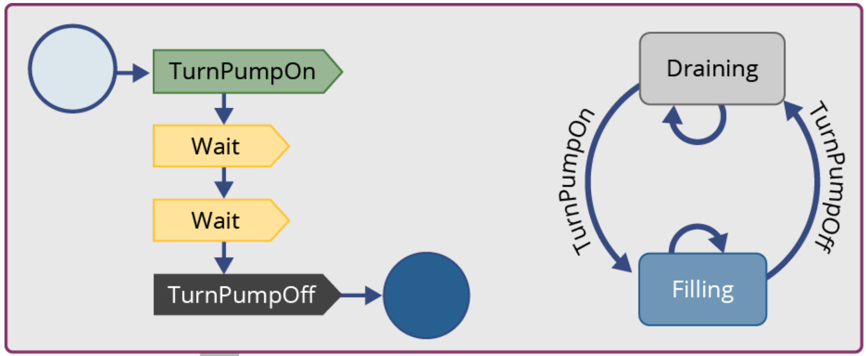
JSON

```
[
  {
    "source": "WaterTank"
    "guideword": "NOT_PROVIDING"
    "controlAction": "TurnPumpOff"
    "context": ["TurnPumpOn", "wait", "wait"]
    "violatedConstraint": "NoOverflow"
  }
]
```

Carnegie Mellon University Software Engineering Institute

Method: Display STPA Output Using RAAML

SysML



RAAML/STPA

	Providing Causes Hazard	Not Providing Causes Hazard
TurnPumpOff		1. TurnPumpOn 2. Wait 3. Wait
TurnPumpOn	1. TurnPumpOn 2. Wait 3. Wait 4. TurnPumpOff	

TLA+

```
Module Tank

Extends Integers

Variables WaterLevel, PumpOn
vars = <<waterLevel,
pumpOn>>

Init ==
   ∧ waterLevel = 0
   ∧ pumpOn = False
TurnPumpOn ==...
```

JSON

```
[
  {
    "goodTrace":    ["TurnPumpOn", "wait", "wait", "TurnPumpOff"]
    "badTrace":     ["TurnPumpOn", "wait", "wait", "wait"]
    "vioComp":      ["WaterTank"]
    "violnv":       ["NoOverflow"]
```

```
      "good Trace":   ["TurnPumpOn", "wait", "wait", "TurnPumpOff"]
      "badTrace":     ["TurnPumpOn", "wait", "wait", "TurnPumpOff", "TurnPumpOn", "wait"]
      "vioComp":      ["WaterTank"]
      "violnv":       ["NoOverflow"]
    }
]
```

JSON

```
[
  {
    "source": "WaterTank"
    "guideword": "NOT_PROVIDING"
    "controlAction": "TurnPumpOff"
    "context": ["TurnPumpOn", "wait", "wait"]
    "violatedConstraint": "NoOverflow"
  }
]
```

Team Contact

Email:
info@sei.cmu.edu

Sam
Procter
PI, Sr. Architecture Researcher

Keaton
Hanna
Associate Software Engineer

Lutz
Wrage
Senior Member of the Technical Staff

Eunsuk
Kang
Associate Professor

Ian
Dardik
Ph.D. Student

Yining
She
Ph.D. Student

For more information, follow this QR code to
https://cmu-soda.github.io/projects/project_fasr.html