# Architecture-Supported Audit Processor Interactive, Query-Driven Assurance

Sam Procter

Jerome Hugues

Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213 [DISTRIBUTION STATEMENT A] Approved for public release and unlimited

distribution.

Carnegie Mellon University Software Engineering Institute

### **Document Markings**

Copyright 2021 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM21-0766

## Outline

#### 1. Background

- 1. AADL / OSATE
- 2. PulseOx Forwarding
- 3. STPA, SAFE
- 2. ASAP: Three Viewpoints
- 3. Future Work

### AADL & OSATE



**Carnegie Mellon University** Software Engineering Institute Architecture-Supported Audit Processor [DISTRIBUTION STATEMENT A] App © 2021 Carnegie Mellon University distribution.

PulseOx\_Forwarding\_System\_imp\_Instance\*

Pulse oximeter reads blood-oxygen saturation from a patient, monitoring software displays an alarm if values are out of expected range

Carnegie Mellon University Software Engineering Institute









• Safety problem to avoid: Incorrect SpO<sub>2</sub> displayed



• Safety problem to avoid: Incorrect SpO<sub>2</sub> displayed



- Safety problem to avoid: Incorrect SpO<sub>2</sub> displayed
- AADL's "Error Modeling" (EMV2) annex can model these error propagations

## **STPA & SAFE**



#### Figure 2.1: Overview of the basic STPA Method

© John Thomas, Nancy Leveson, STPA Handbook, March 2018

https://psas.scripts.mit.edu/home/get\_file.php?name=STPA\_handbook.pdf

Architecture-Supported Audit Processor © 2021 Carnegie Mellon University [DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

# Outline

1. Background

#### 2. ASAP: Three Viewpoints

- 1. Fundamentals
- 2. Connected Neighbors
- 3. Unsafe Control Actions
- 3. Future Work

# **Viewpoint 1: Fundamentals**



#### Figure 2.1: Overview of the basic STPA Method

© John Thomas, Nancy Leveson, STPA Handbook, March 2018

https://psas.scripts.mit.edu/home/get\_file.php?name=STPA\_handbook.pdf



Explanations

Name

Hazardous Factor

System Element

Event Data Port DispSpO2

E

SpO2 Information
BadInfoDisplayed

# Viewpoint 1: Fundamentals (Hierarchy)



Hazard BadInfoDisplayed		
Somantio	Property	Value
Semantic	Hazard BadInfoDisplayed	
	Accident	Accident PatientHarmed
	Constraint	Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	<b>Environment Element</b>	E Abstract patient
	Error Type	Error Type SpO2ValueHigh
	Explanations	LE.
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	Event Data Port DispSpO2

# Viewpoint 1: Fundamentals (Hierarchy)





	Hazard BadInfoDisplayed		
Semantic	Comontio	Property	Value
	Hazard BadInfoDisplayed		
	Accident	Accident PatientHarmed	
	Constraint	Constraint ShowGoodInfo	
	Description	Incorrect information is sent to the display	
	<b>Environment Element</b>	E Abstract patient	
		Error Type	Error Type SpO2ValueHigh
		Explanations	LE
		Hazardous Factor	SpO2 Information
		Name	BadInfoDisplayed

. . . . . .

System Element

Event Data Port DispSpO2



Semantic	Property	Value	
	Hazard BadInfoDisplayed		
	Accident	Accident PatientHarmed	
	Constraint	Constraint ShowGoodInfo	
	Description	Incorrect information is sent to the display	
	<b>Environment Element</b>	E Abstract patient	
	Error Type	Error Type SpO2ValueHigh	
	Explanations	UE:	
		Hazardous Factor	Image: SpO2 Information
		Name	EadInfoDisplayed
		System Element	Event Data Port DispSpO2

How can we tie our rather abstract fundamentals hierarchy to our very concrete system architecture?

SAFE (paraphrased, via STPA) uses the definition of a hazard: a system state, and a worst-case environment state.

# Viewpoint 1: Fundamentals (Link to system)



#### Hazard BadInfoDisplayed

- ···	Property	Value	Hazard = System State + Environment State
Semantic	Hazard BadInfoDisplayed		(Error Type + Port) + (Component)
	Accident	Accident PatientHarmed	
	Constraint	Constraint ShowGoodInfo	Application
	Description	Incorrect information is sent to the display	Application
	Environment Element	E Abstract patient	
	Error Type	Error Type SpO2ValueHigh	
	Explanations	LE .	Sensor Street
	Hazardous Factor	SpO2 Information	Environment
	Name	BadInfoDisplayed	Constraint
	System Element	Event Data Port DispSp02	(Error Type + Port)

# Viewpoint 1: Fundamentals (Link to system)



#### Hazard BadInfoDisplayed

Property value value	Tonnenit State
Semantic ▼Hazard BadInfoDisplayed (Error Type + Port) + (Co	mpo <u>nent)</u>
Accident  Accident PatientHarmed	
Constraint	Application
Description	
Environment Element	/
Error Type	
Explanations	کر ا
Hazardous Factor	invironment H
Name 🗉 BadInfoDisplayedConstraint	
System Element Event Data Port DispSpO2 (Error Type + Port)	

# Viewpoint 1: Fundamentals (Link to system)



#### Hazard BadInfoDisplayed

Property	Value	Hazard = System State + Environment State
Hazard BadInfoDisplayed		(Error Type + Port) + (Component)
Accident	Accident PatientHarmed	
Constraint	Constraint ShowGoodInfo	- Application
Description	Incorrect information is sent to the display	
Environment Element	E Abstract patient	ION
Error Type	Error Type SpO2ValueHigh	
Explanations	LE	Sensor
Hazardous Factor	SpO2 Information	Environment
Name	BadInfoDisplayed	Constraint
System Element	Event Data Port DispSpO2	(Error Type + Port)
	Property <ul> <li>Hazard BadInfoDisplayed</li> <li>Accident</li> <li>Constraint</li> <li>Description</li> <li>Environment Element</li> <li>Error Type</li> <li>Explanations</li> <li>Hazardous Factor</li> <li>Name</li> <li>System Element</li> </ul>	PropertyValueHazard BadInfoDisplayedAccidentAccidentConstraintConstraintDescriptionEnvironment ElementError TypeExplanationsHazardous FactorNameSystem ElementEvent Data Port DispSpO2



#### Figure 2.1: Overview of the basic STPA Method

© John Thomas, Nancy Leveson, STPA Handbook, March 2018

https://psas.scripts.mit.edu/home/get\_file.php?name=STPA\_handbook.pdf









#### **Carnegie Mellon University** Software Engineering Institute



# Viewpoint 3: Unsafe Control Actions



## Interlude: The EMV2 Error Library



# Viewpoint 3: Unsafe Control Actions



**Carnegie Mellon University** Software Engineering Institute

# Outline

- 1. Background
- 2. ASAP: Three Viewpoints
- 3. Future Work

# Future Work

- 1. The "Focus" Action
- 2. Discovering accident causation