



Architecture-Supported Audit Processor

Interactive, Query-Driven Assurance

Sam Procter

Jerome Hugues

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

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

The screenshot displays the Eclipse IDE interface for AADL development. The top-left pane shows the project tree for 'Pulse0x_Forwarding_System_Impl_Interface.aad'. The top-right pane shows a UML diagram of the 'Pulse0x_Forwarding_System_Impl_Interface' component, including its internal structure and connections to other components. The bottom-left pane shows the source code for 'Pulse0x_Interface.aad', which defines the component's interface and properties. The bottom-right pane shows the 'Outline' view with a tree structure of the project's components.

```
package Pulse0x_Interface
export
  MWP_Errors, Pulse0x_Forwarding_Errors, Pulse0x_Forwarding_Types;
with MWP_Properties;

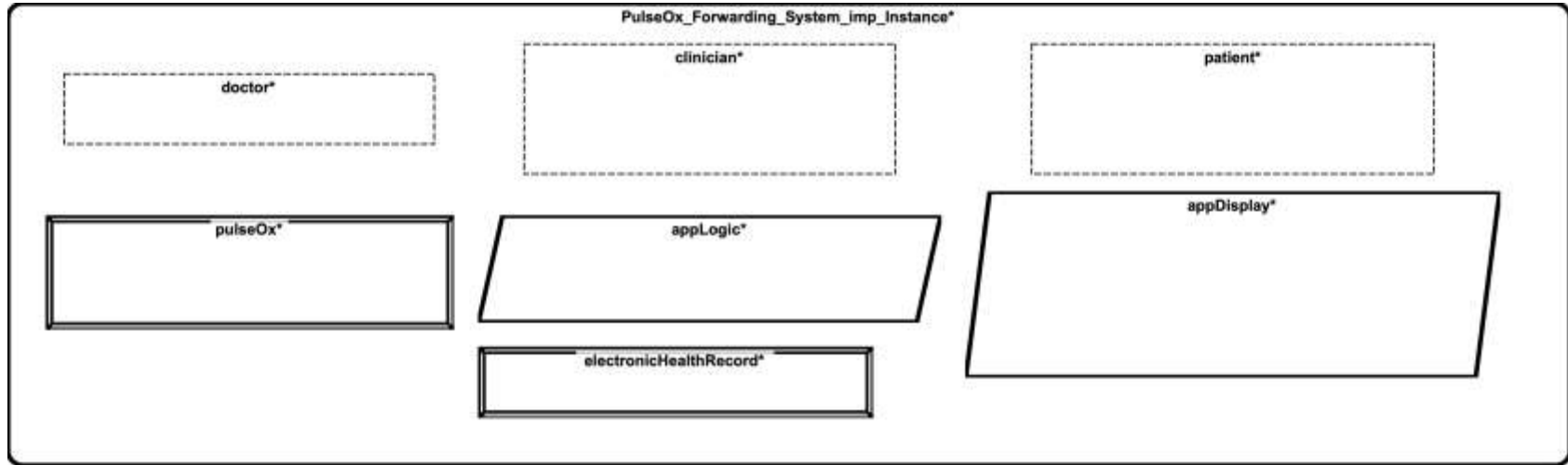
device if(EpoInterface
  features
    SensorInput: in Feature;
    P0OutSp02: out sensor auto out: Pulse0x_Forwarding_Types::Sp02 {
      MWP_Properties::Exchange_Name => "sp02_out"};
  flows
    Flow0x: flow sensor P0OutSp02;
  properties
    MWP_Properties::Component_Type == sensor;
  error ENV2 {+
    use types Pulse0x_Forwarding_Errors - MWP_Errors;
  }
end if(EpoInterface);
```

The PulseOx Forwarding Example

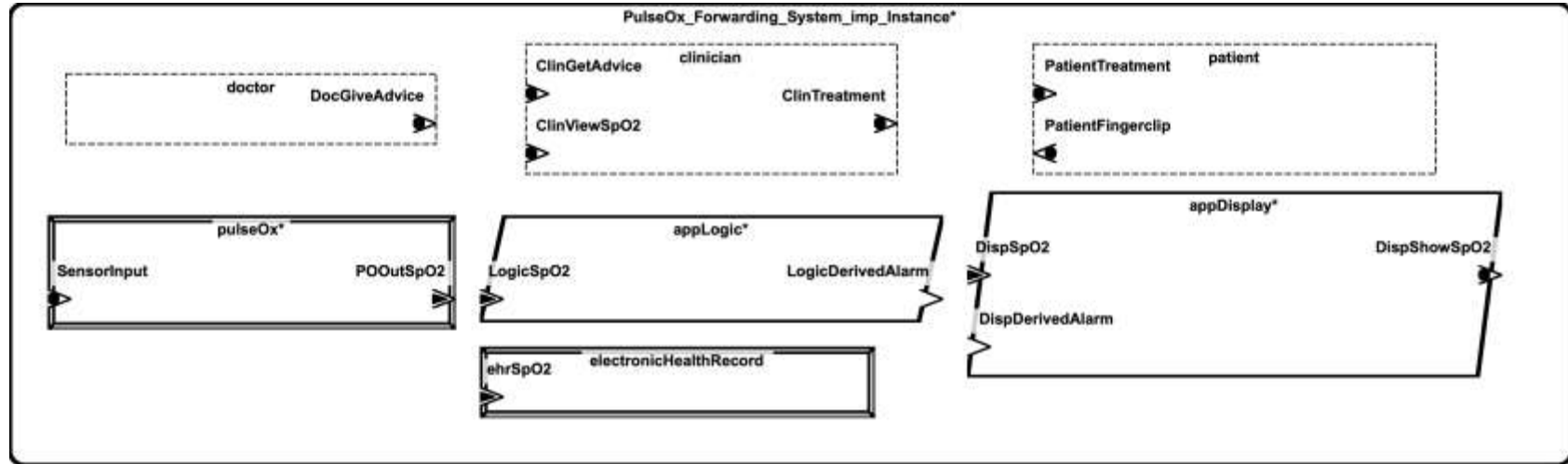
PulseOx_Forwarding_System_imp_Instance*

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

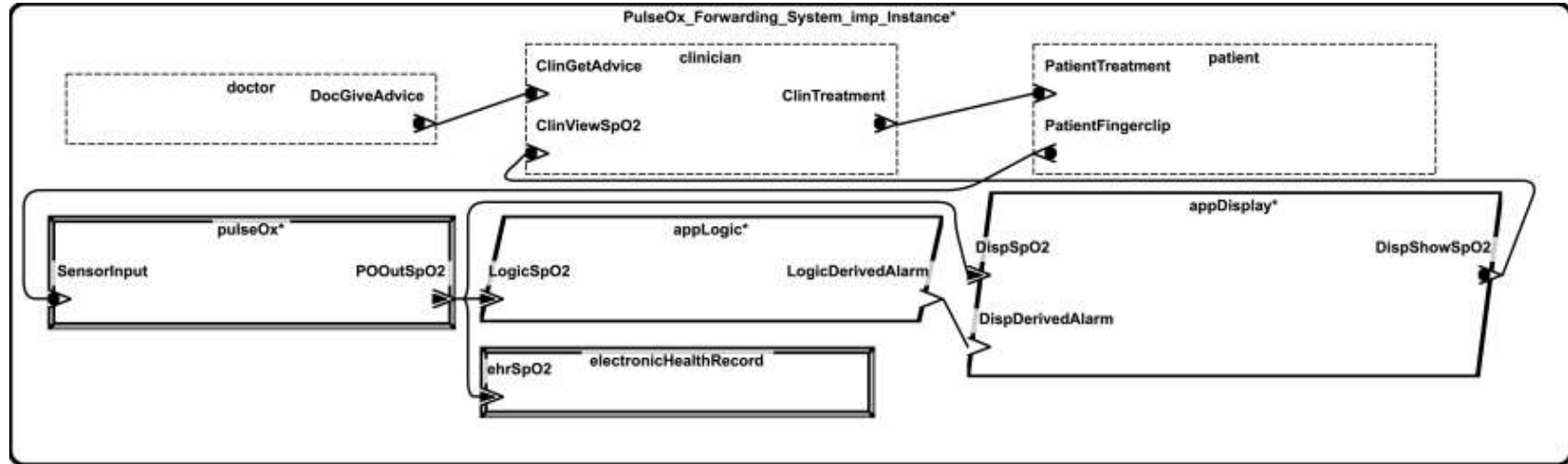
The PulseOx Forwarding Example



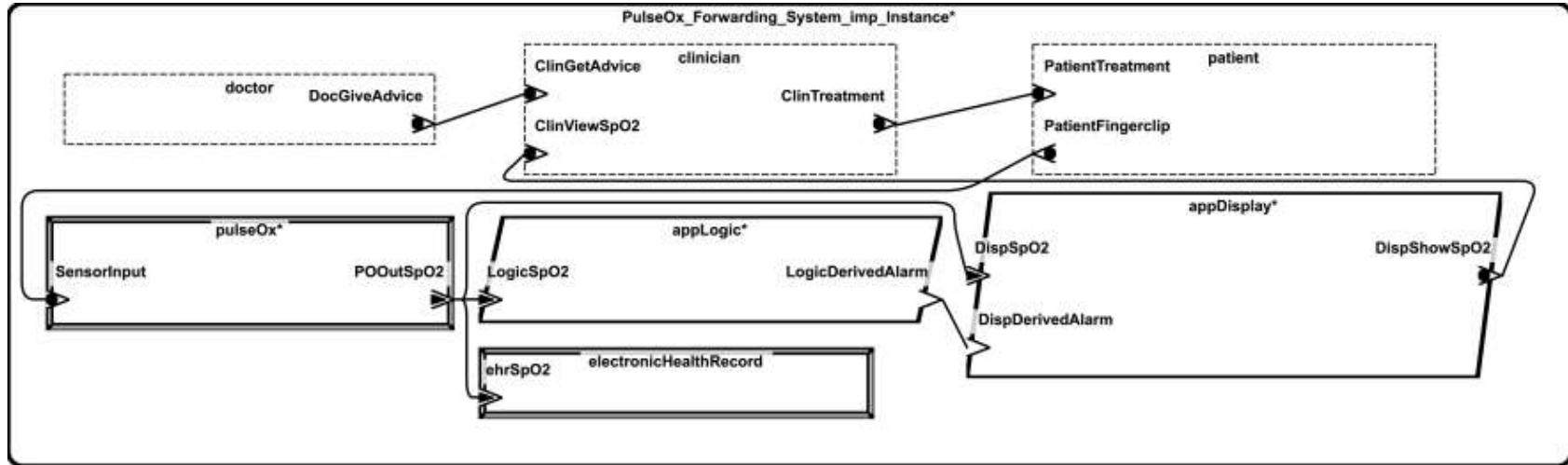
The PulseOx Forwarding Example



The PulseOx Forwarding Example

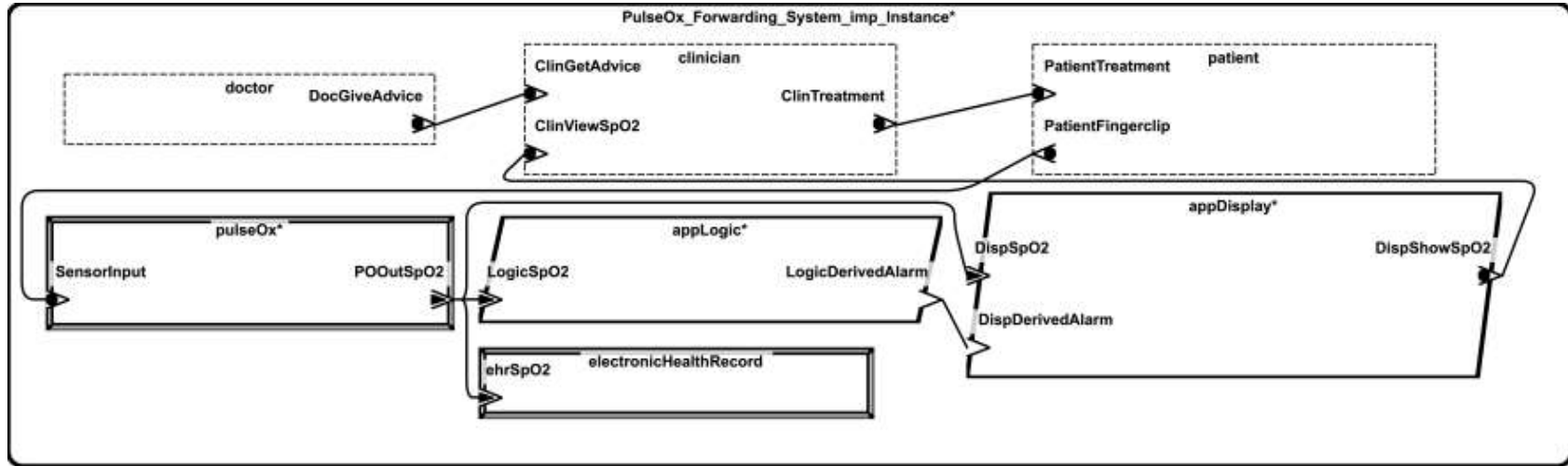


The PulseOx Forwarding Example



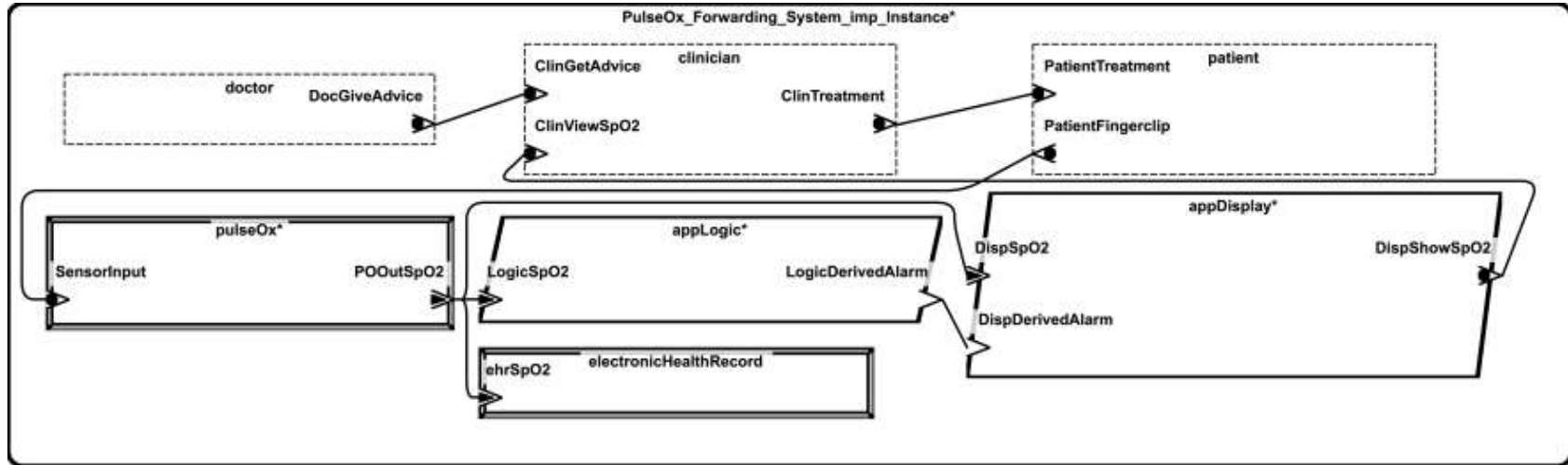
- Safety problem to avoid: Incorrect SpO₂ displayed

The PulseOx Forwarding Example



- Safety problem to avoid: Incorrect SpO₂ displayed

The PulseOx Forwarding Example



- Safety problem to avoid: Incorrect SpO₂ displayed
- AADL's "Error Modeling" (EMV2) annex can model these error propagations

STPA & SAFE

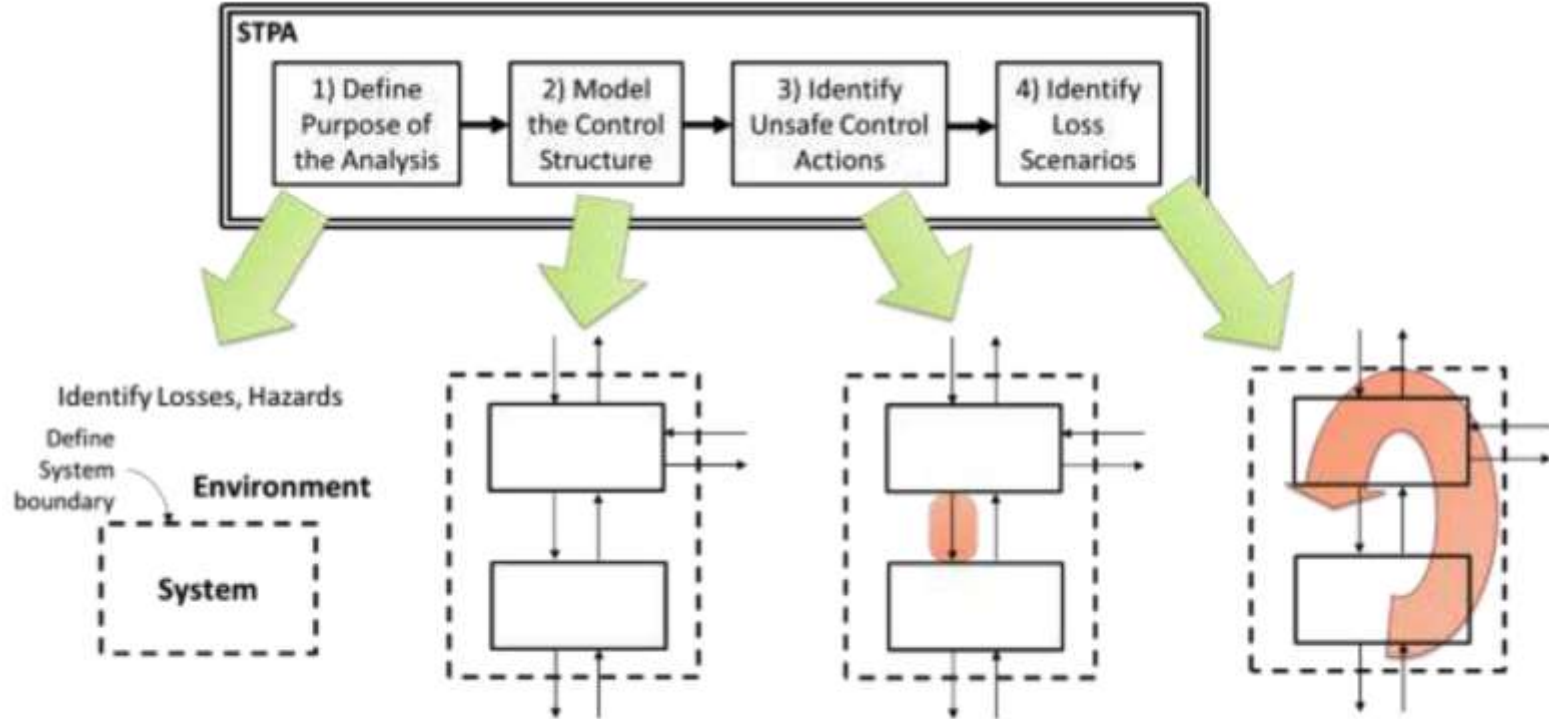


Figure 2.1: Overview of the basic STPA Method

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

https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

Outline

1. Background
- 2. ASAP: Three Viewpoints**
 1. Fundamentals
 2. Connected Neighbors
 3. Unsafe Control Actions
3. Future Work

Viewpoint 1: Fundamentals

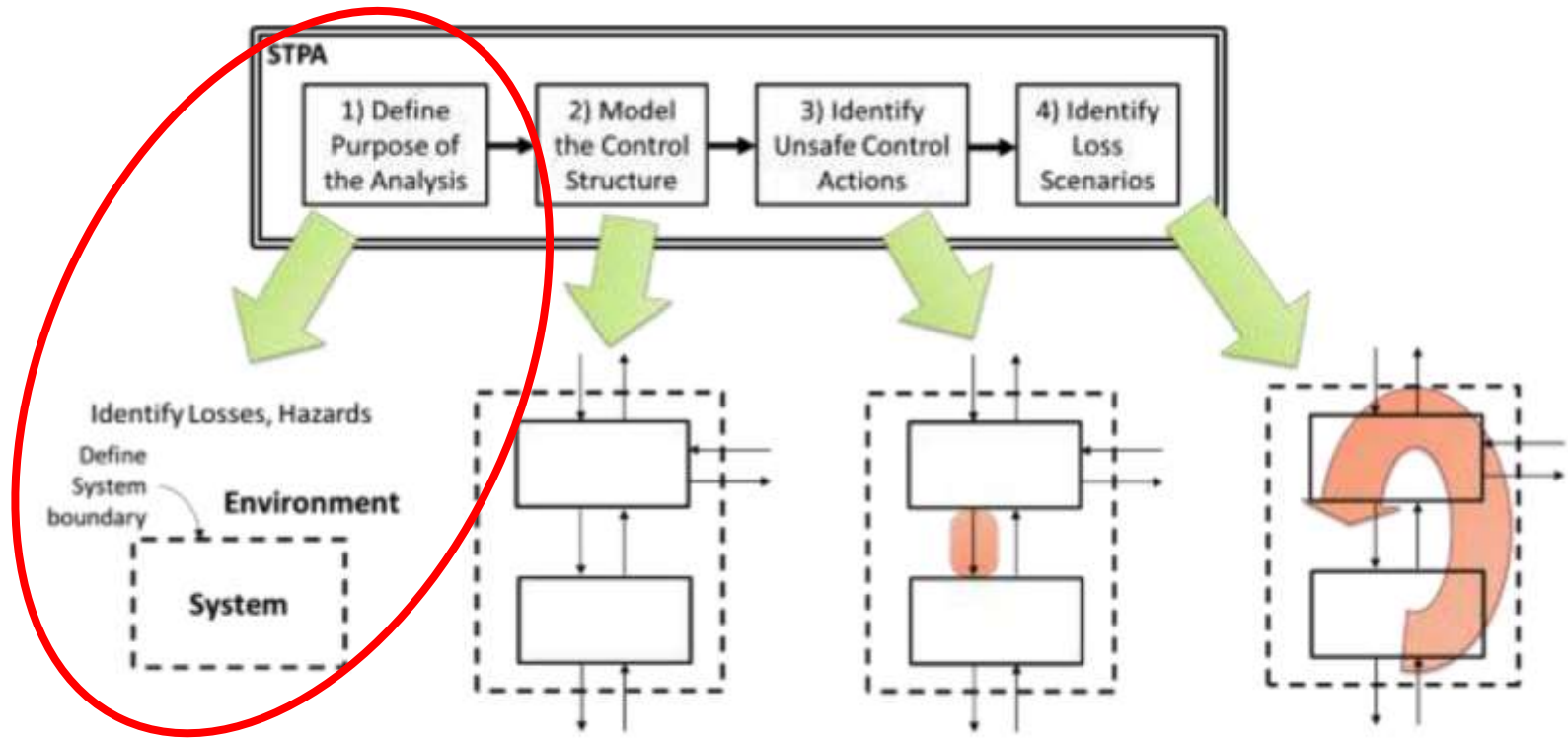


Figure 2.1: Overview of the basic STPA Method

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

https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

Viewpoint 1: Fundamentals (Hierarchy)

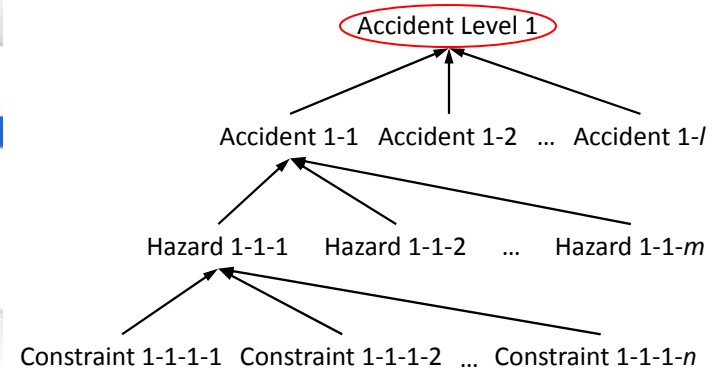
representations.aird pulseox-forwarding.safe2 new Fundamentals

- DeathOrInjury
 - PatientHarmed
 - BadInfoDisplayed
 - ShowGoodInfo
 - InfoLate
 - ShowInfoOnTime

Problems Properties AADL Prop Classifier In Progress Error Log

Hazard BadInfoDisplayed

Semantic	Property	Value
	▼ Hazard BadInfoDisplayed	
	Accident	◆ Accident PatientHarmed
	Constraint	◆ Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	Environment Element	Abstract patient
	Error Type	◆ Error Type SpO2ValueHigh
	Explanations	
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	▶ Event Data Port DispSpO2



Viewpoint 1: Fundamentals (Hierarchy)

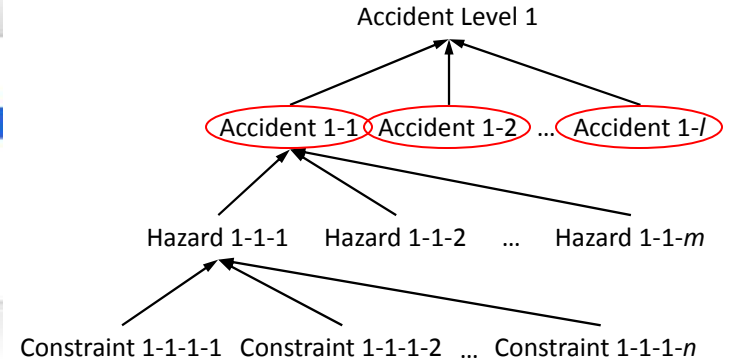
representations.aird pulseox-forwarding.safe2 new Fundamentals

- DeathOrInjury
 - PatentHarmed**
 - BadInfoDisplayed
 - ShowGoodInfo
 - InfoLate
 - ShowInfoOnTime

Problems Properties AADL Prop Classifier In Progress Error Log

Hazard BadInfoDisplayed

Semantic	Property	Value
	▼ Hazard BadInfoDisplayed	
	Accident	◆ Accident PatentHarmed
	Constraint	◆ Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	Environment Element	Abstract patient
	Error Type	◆ Error Type SpO2ValueHigh
	Explanations	
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	▶ Event Data Port DispSpO2

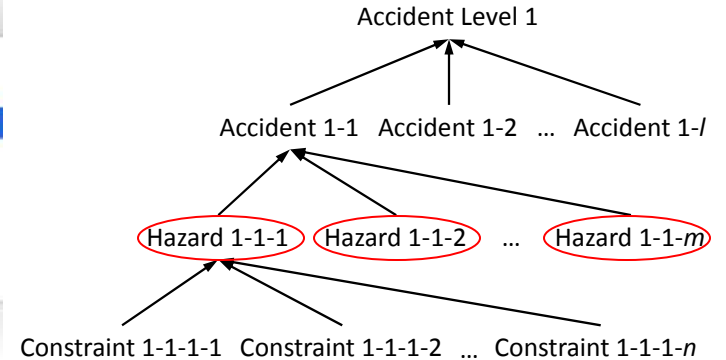


Viewpoint 1: Fundamentals (Hierarchy)

The screenshot shows a software interface with a project tree on the left and a toolbar at the bottom. The project tree is expanded to show a hierarchy: **DeathOrInjury** > **PatientHarmed** > **BadInfoDisplayed** (highlighted in blue) > **InfoLate** (circled in red). Other items in the tree include **ShowGoodInfo** and **ShowInfoOnTime**. The toolbar contains icons for **Problems**, **Properties**, **AADL Prop**, **Classifier In**, **Progress**, and **Error Log**.

Hazard BadInfoDisplayed

Semantic	Property	Value
	▼ Hazard BadInfoDisplayed	
	Accident	◆ Accident PatientHarmed
	Constraint	◆ Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	Environment Element	Abstract patient
	Error Type	◆ Error Type SpO2ValueHigh
	Explanations	
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	▶ Event Data Port DispSpO2



Viewpoint 1: Fundamentals (Hierarchy)

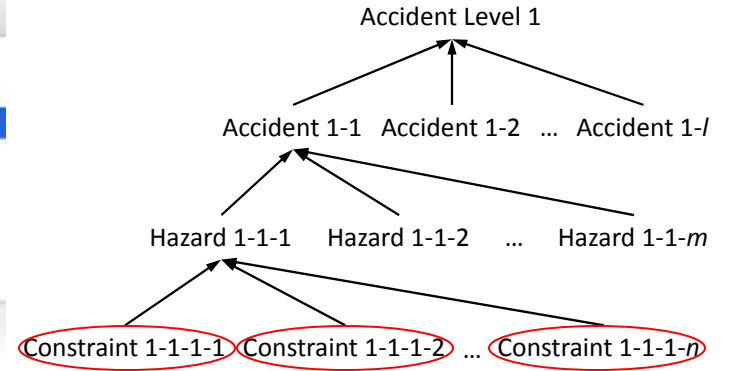
representations.aird pulseox-forwarding.safe2 new Fundamentals

- DeathOrInjury
 - PatientHarmed
 - BadInfoDisplayed
 - ShowGoodInfo
 - InfoLate
 - ShowInfoOnTime

Problems Properties AADL Prop Classifier In Progress Error Log

Hazard BadInfoDisplayed

Semantic	Property	Value
	▼ Hazard BadInfoDisplayed	
	Accident	◆ Accident PatientHarmed
	Constraint	◆ Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	Environment Element	Abstract patient
	Error Type	◆ Error Type SpO2ValueHigh
	Explanations	
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	▶ Event Data Port DispSpO2

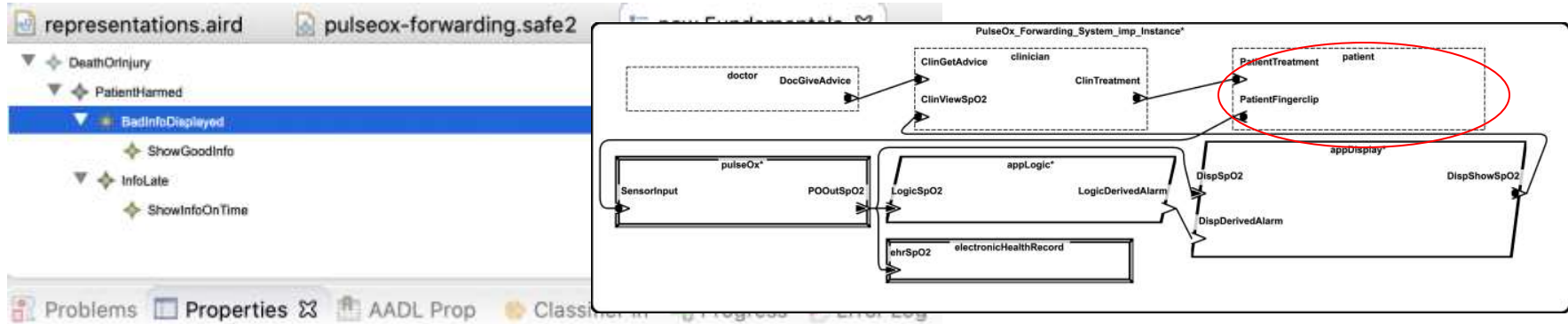


Viewpoint 1: Fundamentals

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.

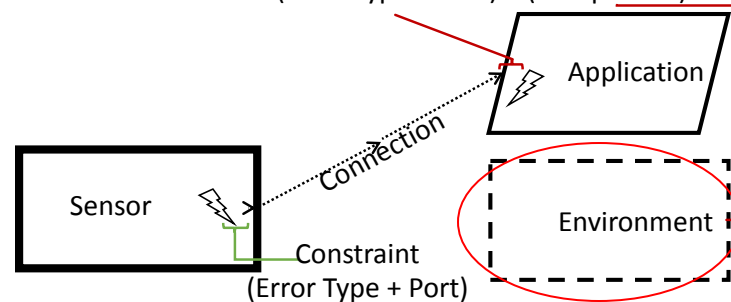
Viewpoint 1: Fundamentals (Link to system)



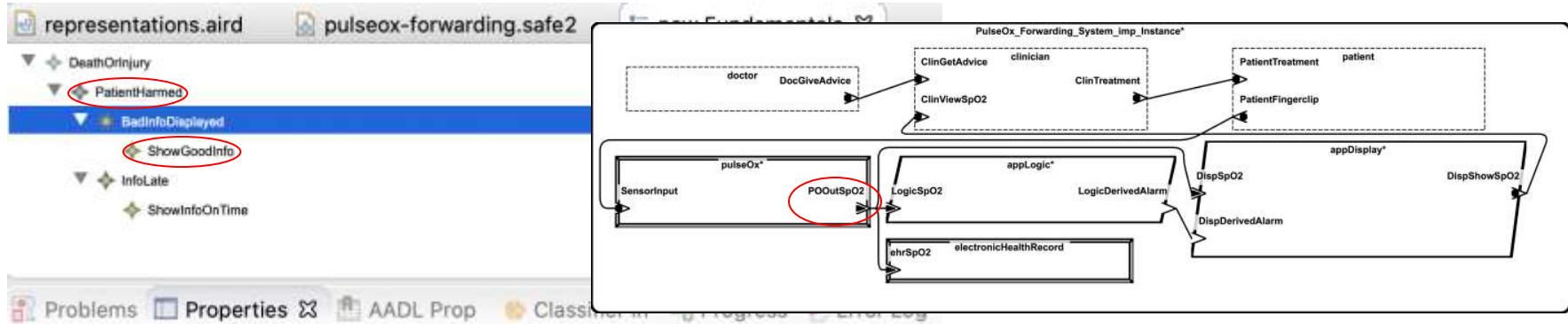
⚡ Hazard BadInfoDisplayed

Semantic	Property	Value
	▼ Hazard BadInfoDisplayed	
	Accident	⚡ Accident PatientHarmed
	Constraint	⚡ Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	Environment Element	⚡ Abstract patient
	Error Type	⚡ Error Type SpO2ValueHigh
	Explanations	
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	▶ Event Data Port DispSpO2

Hazard = System State + Environment State
(Error Type + Port) + (Component)



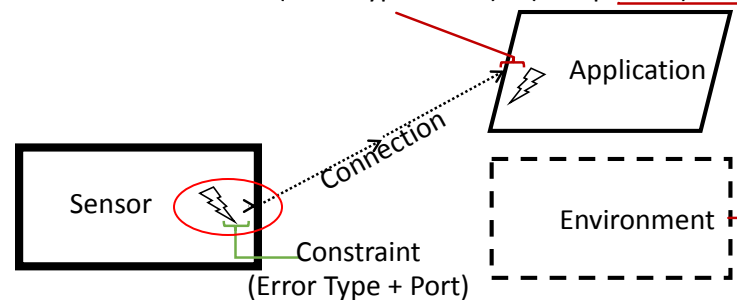
Viewpoint 1: Fundamentals (Link to system)



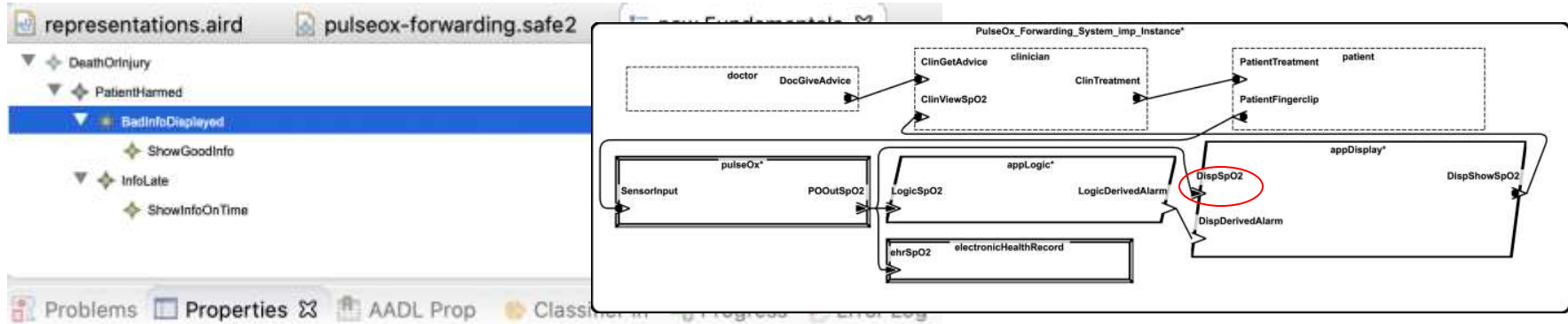
Hazard BadInfoDisplayed

Semantic	Property	Value
	▼ Hazard BadInfoDisplayed	
	Accident	Accident PatientHarmed
	Constraint	Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	Environment Element	Abstract patient
	Error Type	Error Type SpO2ValueHigh
	Explanations	
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	Event Data Port DispSpO2

Hazard = System State + Environment State
(Error Type + Port) + (Component)



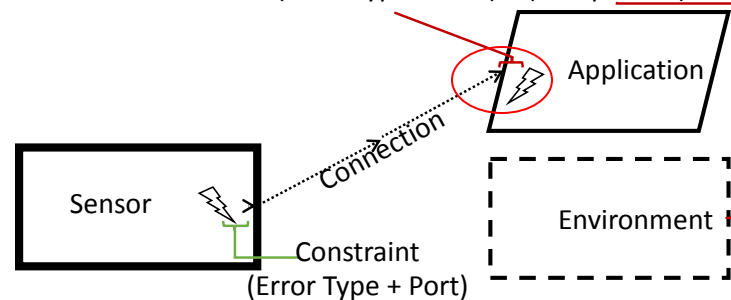
Viewpoint 1: Fundamentals (Link to system)



Hazard BadInfoDisplayed

Semantic	Property	Value
	▼ Hazard BadInfoDisplayed	
	Accident	◆ Accident PatientHarmed
	Constraint	◆ Constraint ShowGoodInfo
	Description	Incorrect information is sent to the display
	Environment Element	Abstract patient
	Error Type	◆ Error Type SpO2ValueHigh
	Explanations	
	Hazardous Factor	SpO2 Information
	Name	BadInfoDisplayed
	System Element	▶ Event Data Port DispSpO2

Hazard = System State + Environment State
(Error Type + Port) + (Component)



Viewpoint 2: Connected Neighbors

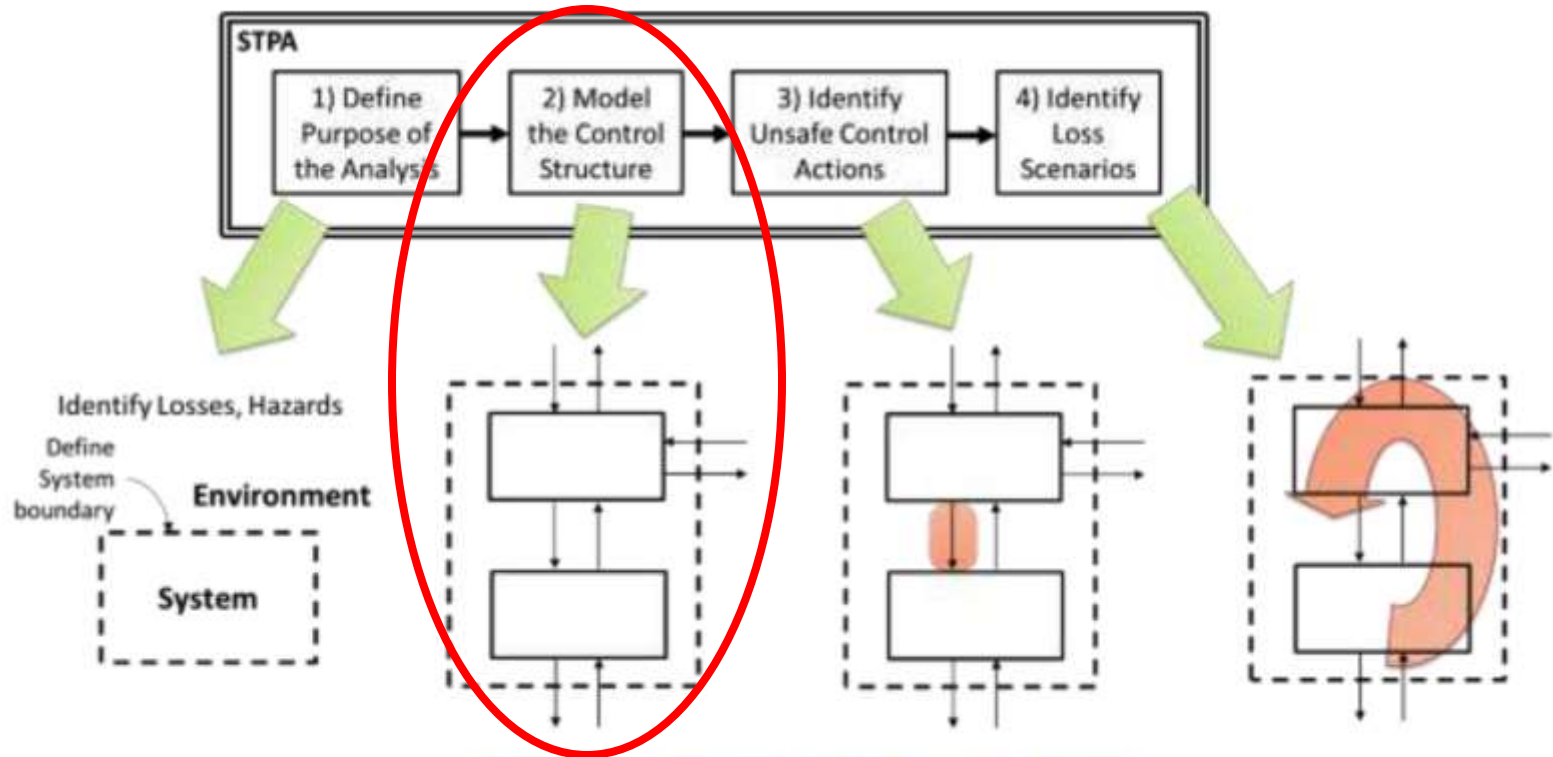
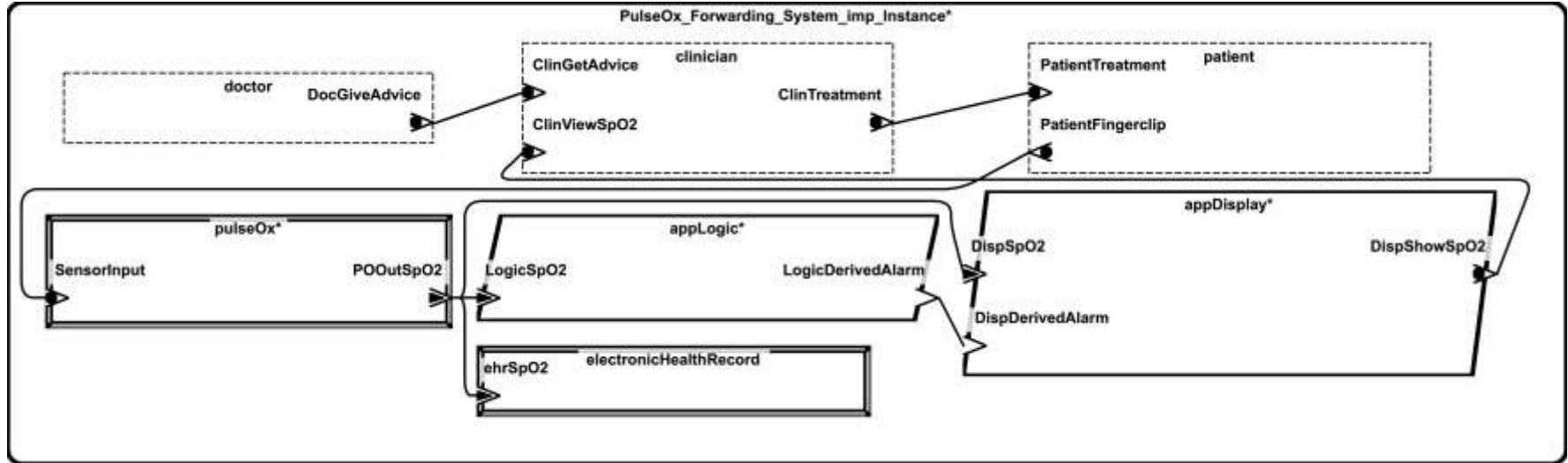


Figure 2.1: Overview of the basic STPA Method

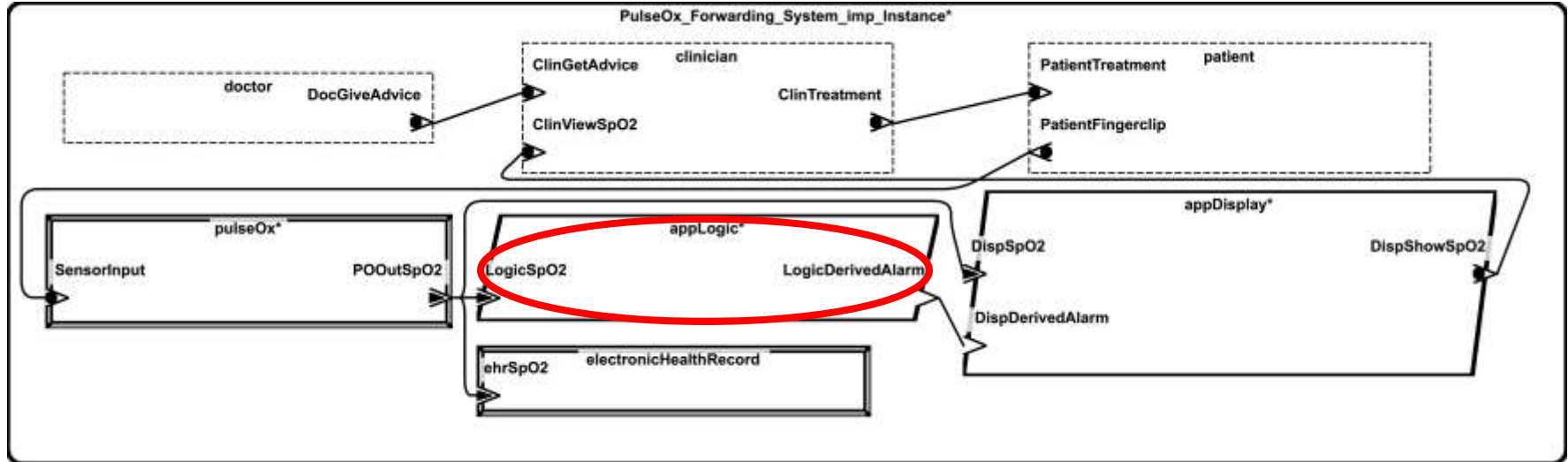
© John Thomas, Nancy Leveson, STPA Handbook, March 2018

https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

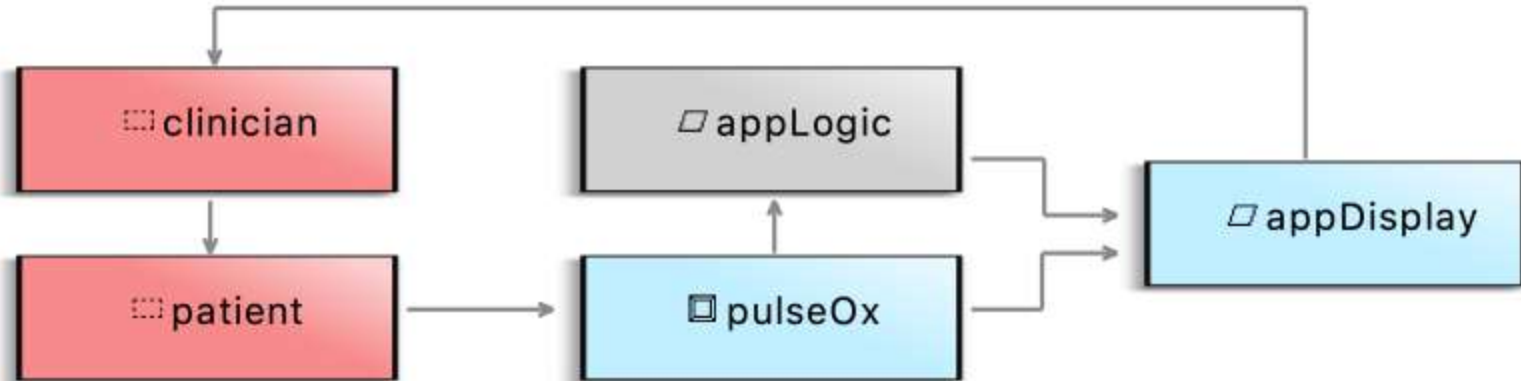
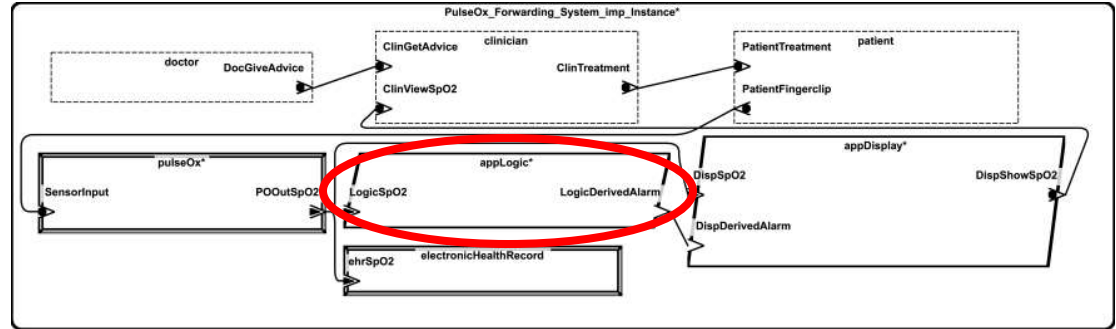
Viewpoint 2: Connected Neighbors



Viewpoint 2: Connected Neighbors



Viewpoint 2: Connected Neighbors



Viewpoint 3: Unsafe Control Actions

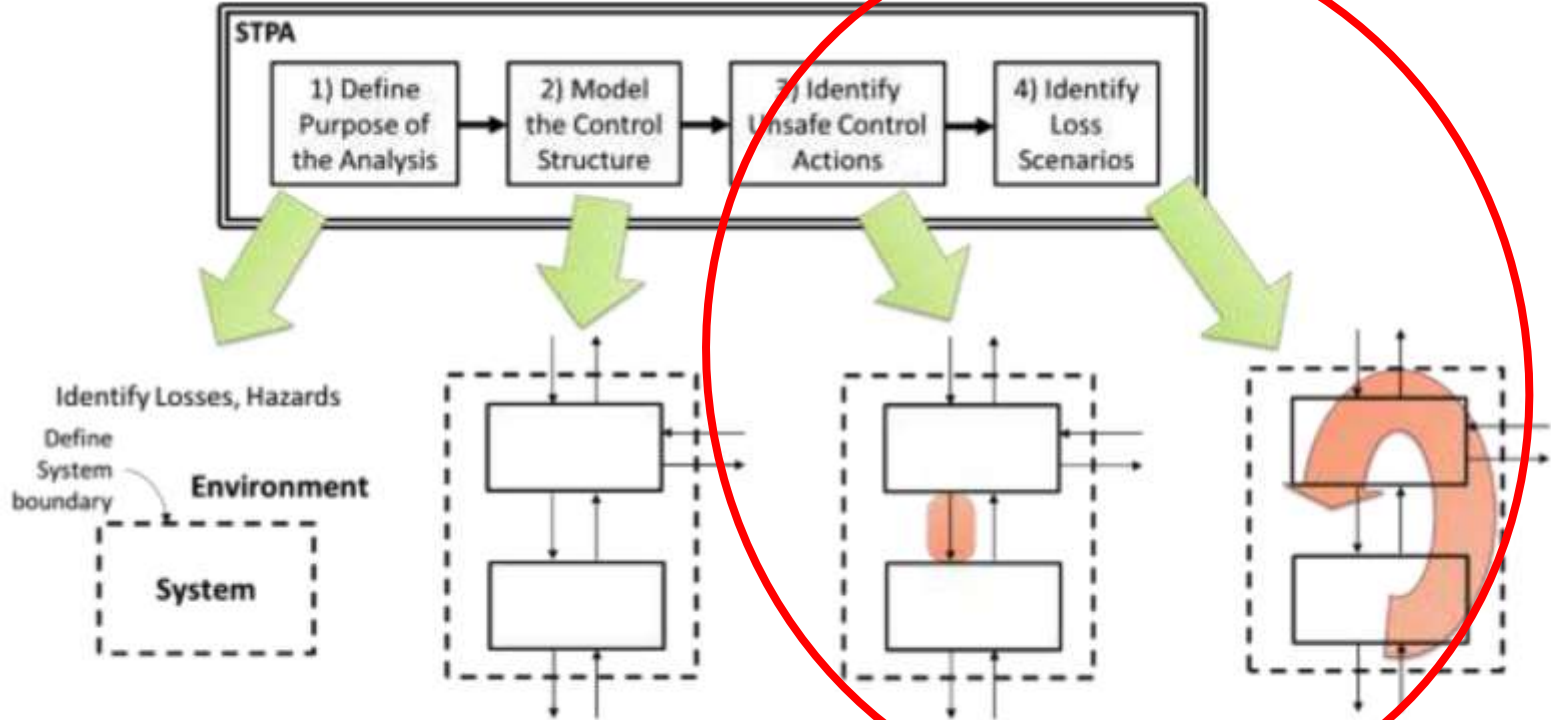
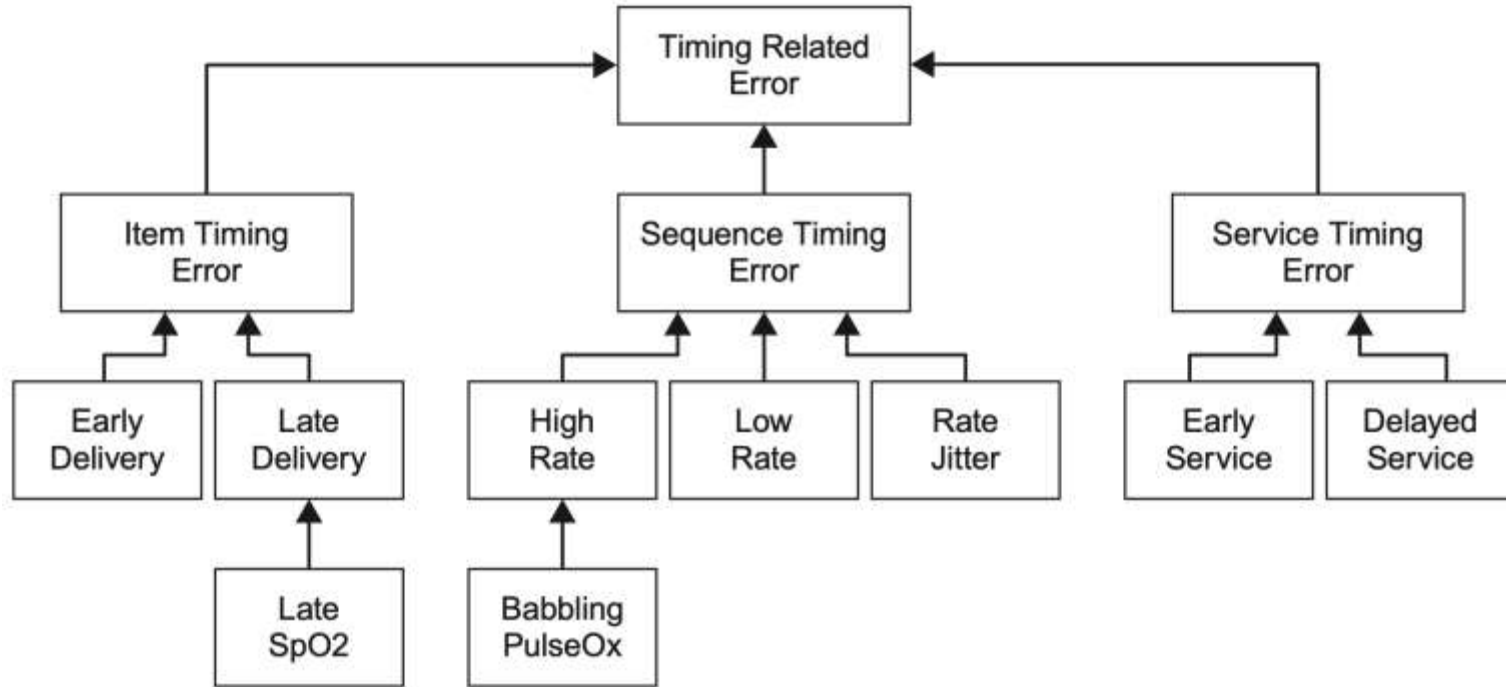


Figure 2.1: Overview of the basic STPA Method

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

https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

Interlude: The EMV2 Error Library



Viewpoint 3: Unsafe Control Actions

Communication Channels
(ie, control actions and
sensor feedback)

- ◆ patient.PatientFingerclip -> pulseOx.SensorInput
- ◆ pulseOx.POOOutSpO2 -> electronicHealthRecord.ehrSpO2
- ◆ doctor.DocGiveAdvice -> clinician.ClinGetAdvice
- ◆ pulseOx.POOOutSpO2 -> appLogic.StoreSpO2Thread.incoming_spo2
- ◆ appDisplay.DispShowSpO2 -> clinician.ClinViewSpO2
- ◆ clinician.ClinTreatment -> patient.PatientTreatment
- ◆ appLogic.CheckSpO2Thread.Alarm -> appDisplay.HandleAlarmThread.Ala...
- ◆ pulseOx.POOOutSpO2 -> appDisplay.UpdateSpO2Thread.SpO2

Top-Level Errors
(ie, abstract guidewords)

◆ ItemValueError ◆ ItemTimingError ◆ ViolatedConstraint ◆ ServiceError

X

X

X

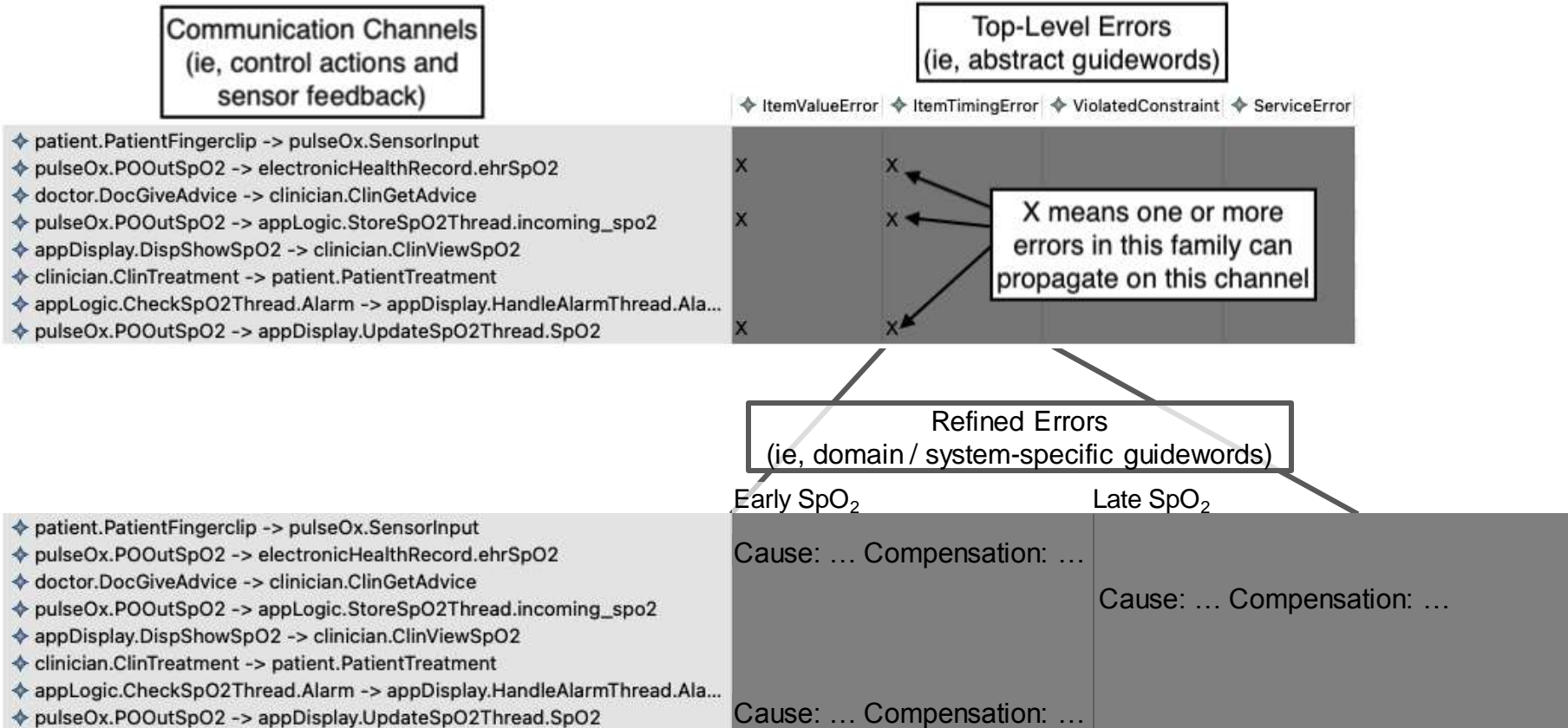
X

X

X

X means one or more
errors in this family can
propagate on this channel

Viewpoint 3: Unsafe Control Actions



Outline

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

Future Work

1. The “Focus” Action
2. Discovering accident causation