An Architecturally-Integrated, Systems-Based Hazard Analysis for Medical Applications

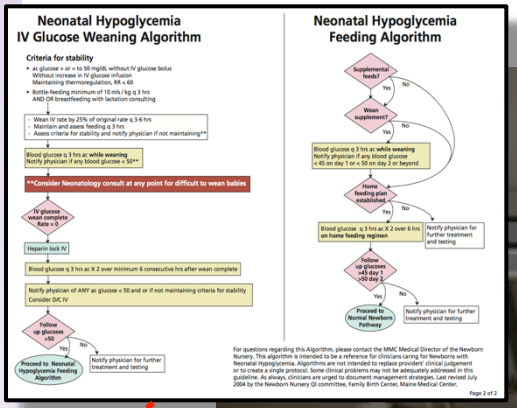
http://cis.ksu.edu/~samprocter

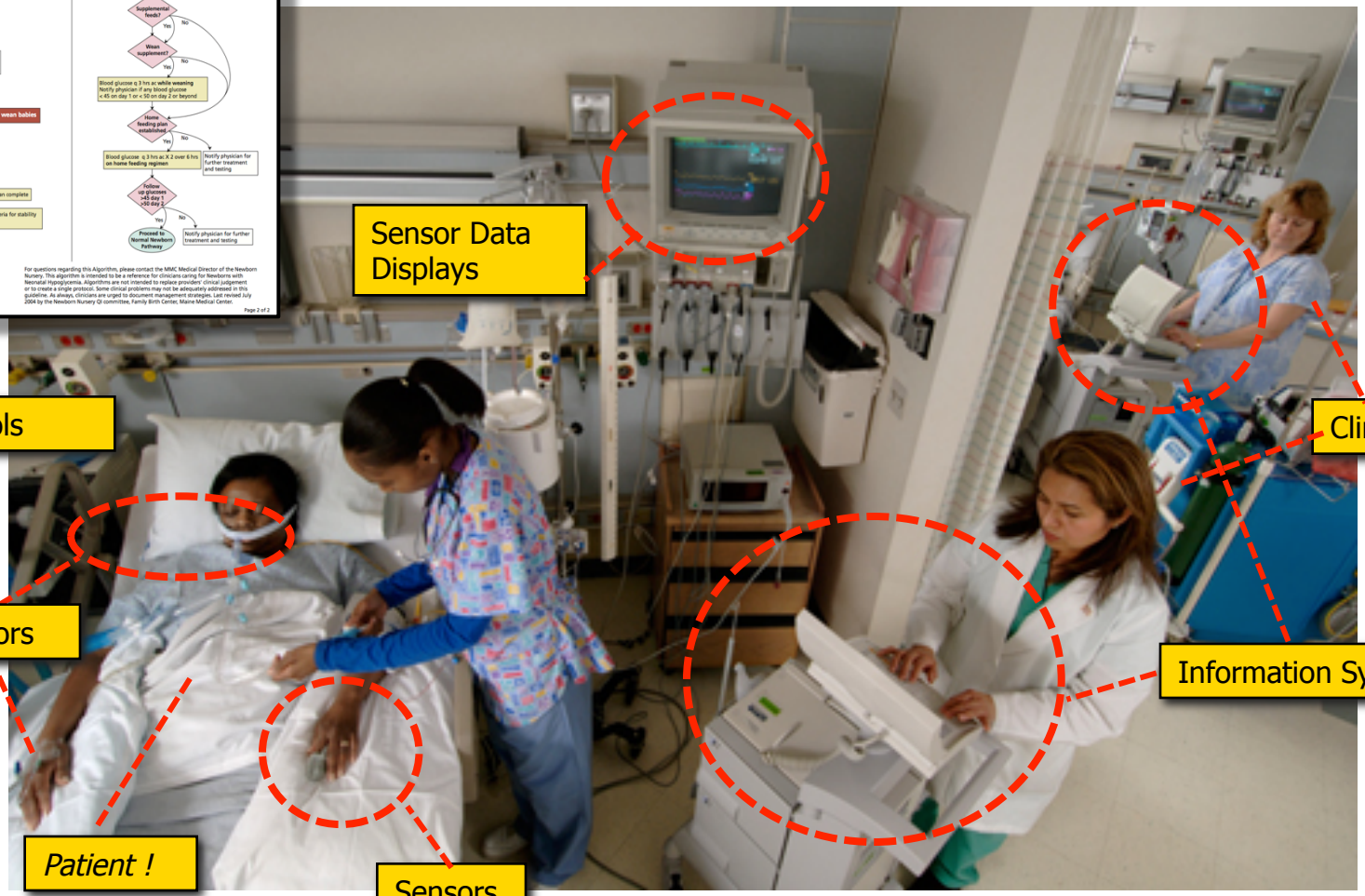
Sam Procter and John Hatcliff
SAnToS Lab
Kansas State University

Support:

This work is supported in part by the US National Science Foundation (NSF) (#1239543), the NSF US Food and Drug Administration Scholar-in-Residence Program (#1355778) and the National Institutes of Health / NIBIB Quantum Program.

Health Care Involves A Variety of System Components





Clinical Protocols

Sensor Data Displays

Clinicians

Actuators

Information Systems

Patient !

Sensors

Outline

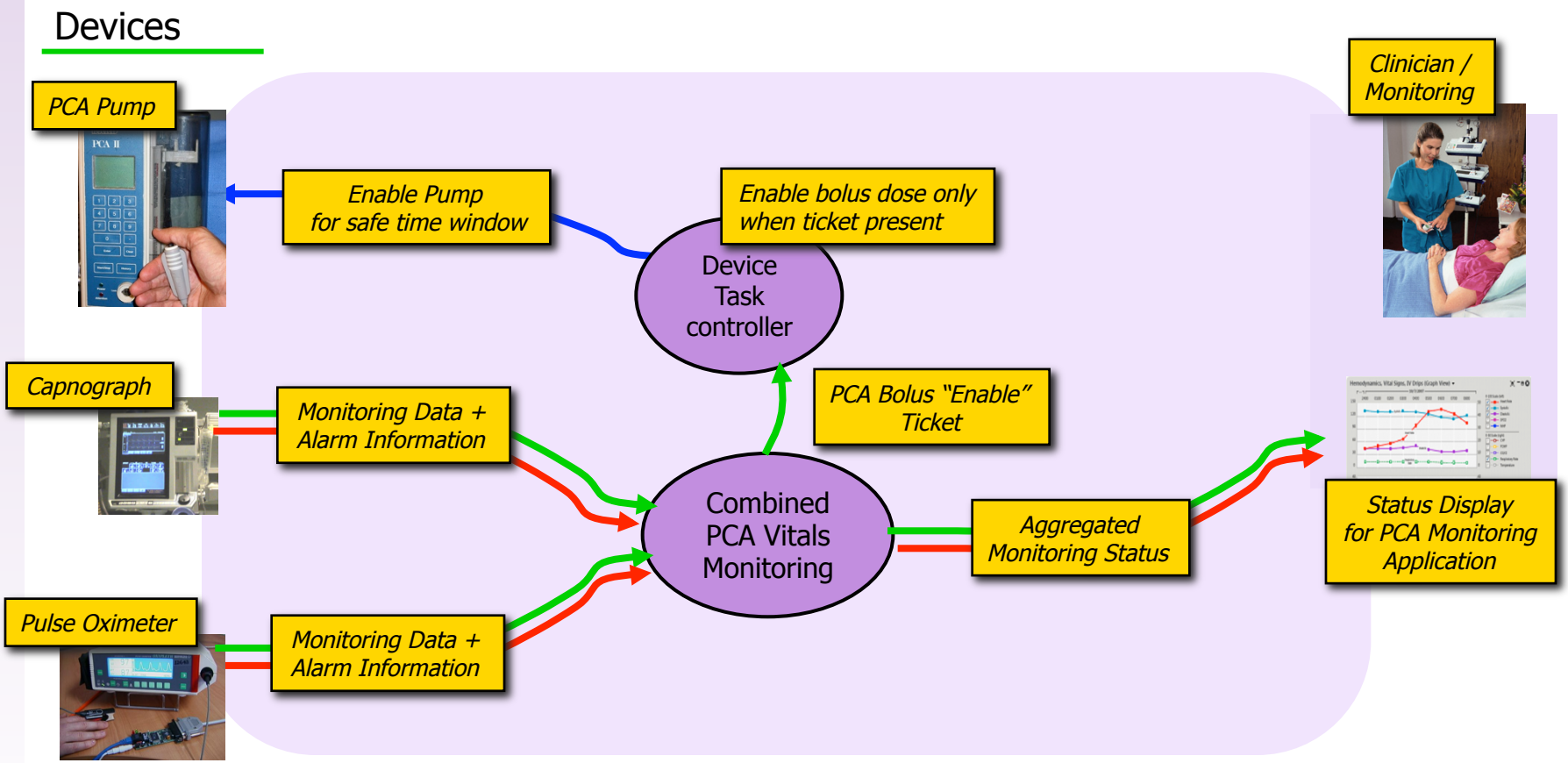
- Motivation
- Report
  - Annotations
  - Generation
- Language
- Impacts

PCA Interlock Scenario

- Patients are commonly given patient-controlled analgesics after surgery
- Crucial to care, but numerous issues related to safety
- Data for disabling the pump exists now (just a system invariant) -- we just need to integrate it

PCA Pump Safety Interlock

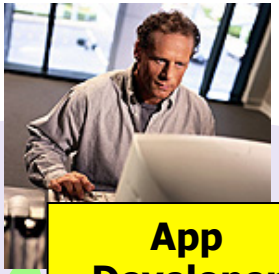
Fully leverage device data streams and the ability to *control* devices



Vision

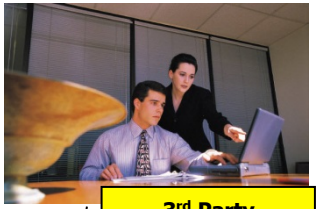
Analyses and Regulatory Artifacts

Clinical Use Case / Workflow Description

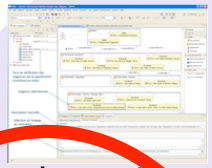


App Developer

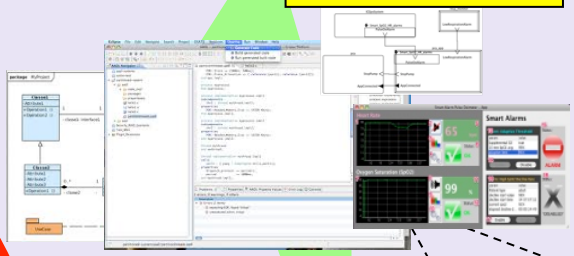
Assurance Case



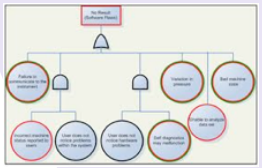
3rd Party Certifiers



Requirements

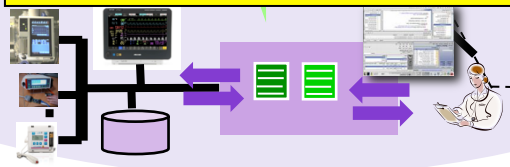


3rd Party
ICE Conformance
& Safety Certification
Submission Package



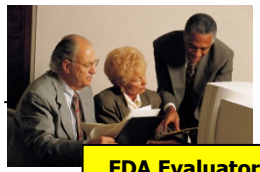
Hazard Analysis

Medical Application Platform



App Deployment

FDA 510K
Submission Package

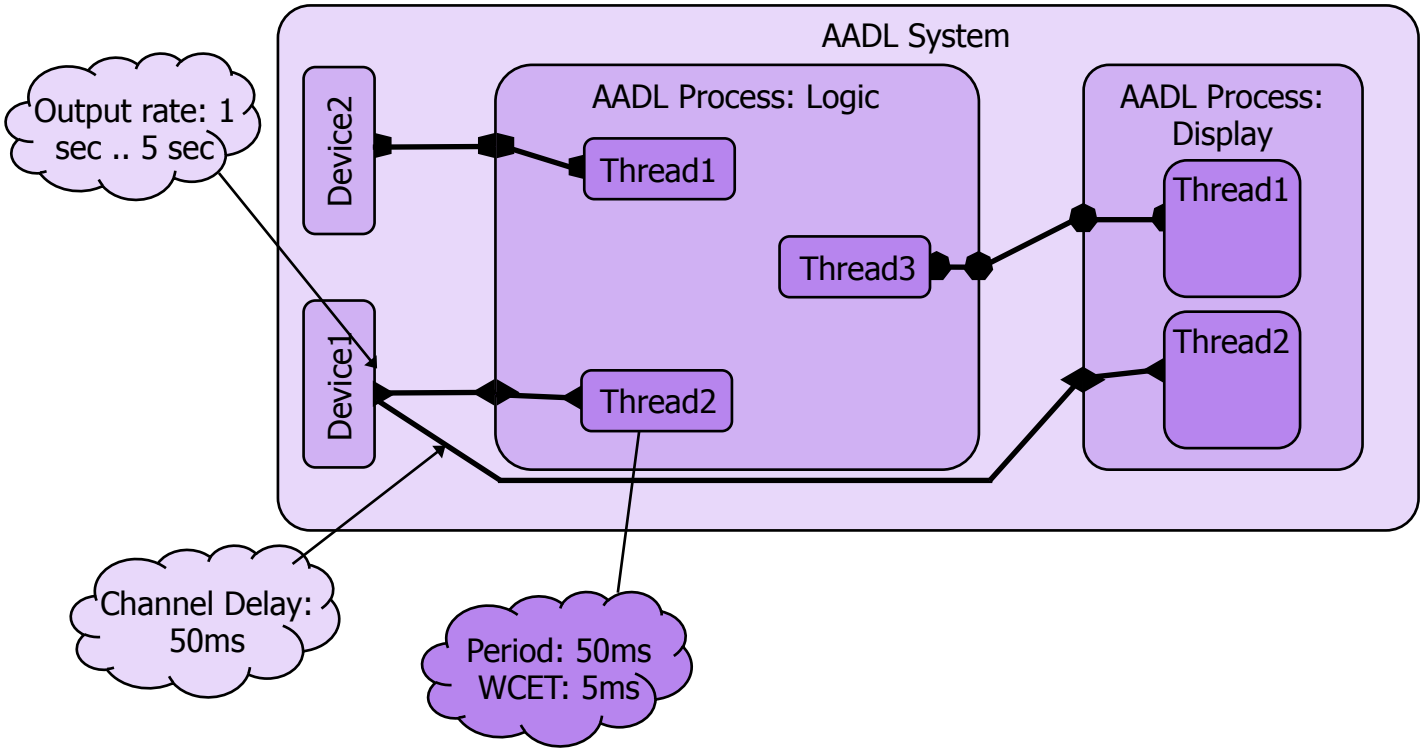


FDA Evaluators

Risk Assessment

Language

Model



STPA

Fundamentals

- Fundamentals
  - Accident Levels
  - Accidents
  - System Boundaries
  - Hazards
  - Safety Constraints
  - Control Actions
  - Control Structure

Example

1. An inadvertent "Pump Normally" command is sent to the pump [PatientHarmed]

2. Commands are sent to the pump too quickly [PCADamage]

```
InadvertentPumpNormally : constant MAP_Error_Properties::Hazard => [
        Number => 1;
        Description => "An inadvertent `Pump Normally` command is sent to the pump.";
        Accident => PulseOx_Forwarding_Error_Properties::PatientHarmed;
];
```

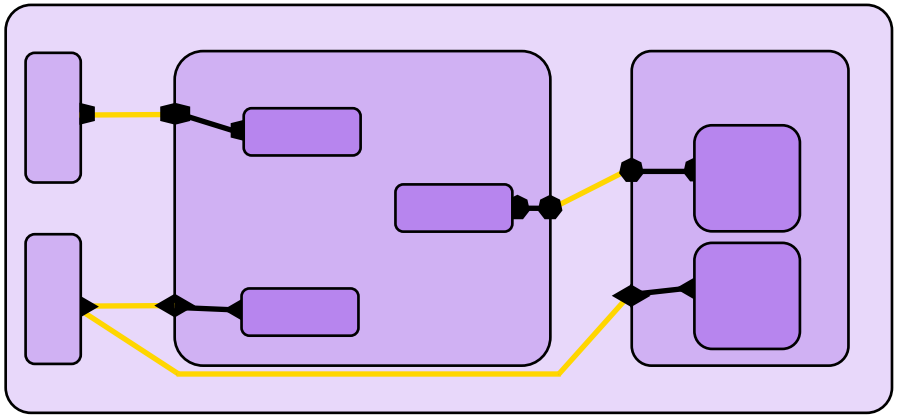
STPA

Fundamentals

- Fundamentals
  - Accident Levels
  - Accidents
  - System Boundaries
  - Hazards
  - Safety Constraints
  - Control Actions
  - Control Structure

Example

1. App -> Pump: Pump Normally
2. PulseOx -> App[1]: SpO_2 = 95
3. App -> Display: Patient = Ok



1: Also referred to as "Feedback"

STPA

Step 1: Identifying Potentially Hazardous Control Actions

- Hazardous Control Actions
  - Cross-product of control actions and STPA guidewords

Control Action	Providing	Not Providing	Applied too Long	Stopped too Soon	Early	Late
App -> Pump: Pump Normally	PH	Not Hazardous	PH	Not Hazardous	PH	Not Hazardous
App -> Disp: Patient Ok	BID	BID	BID	BID	BID	BID
PulseOx->App: Provide SpO_2	Not Hazardous	PH, BID	Not Hazardous	PH, BID	Not Hazardous	PH, BID
PulseOx->App: Provide Pulse Rate	Not Hazardous	PH, BID	Not Hazardous	PH, BID	Not Hazardous	PH, BID

STPA

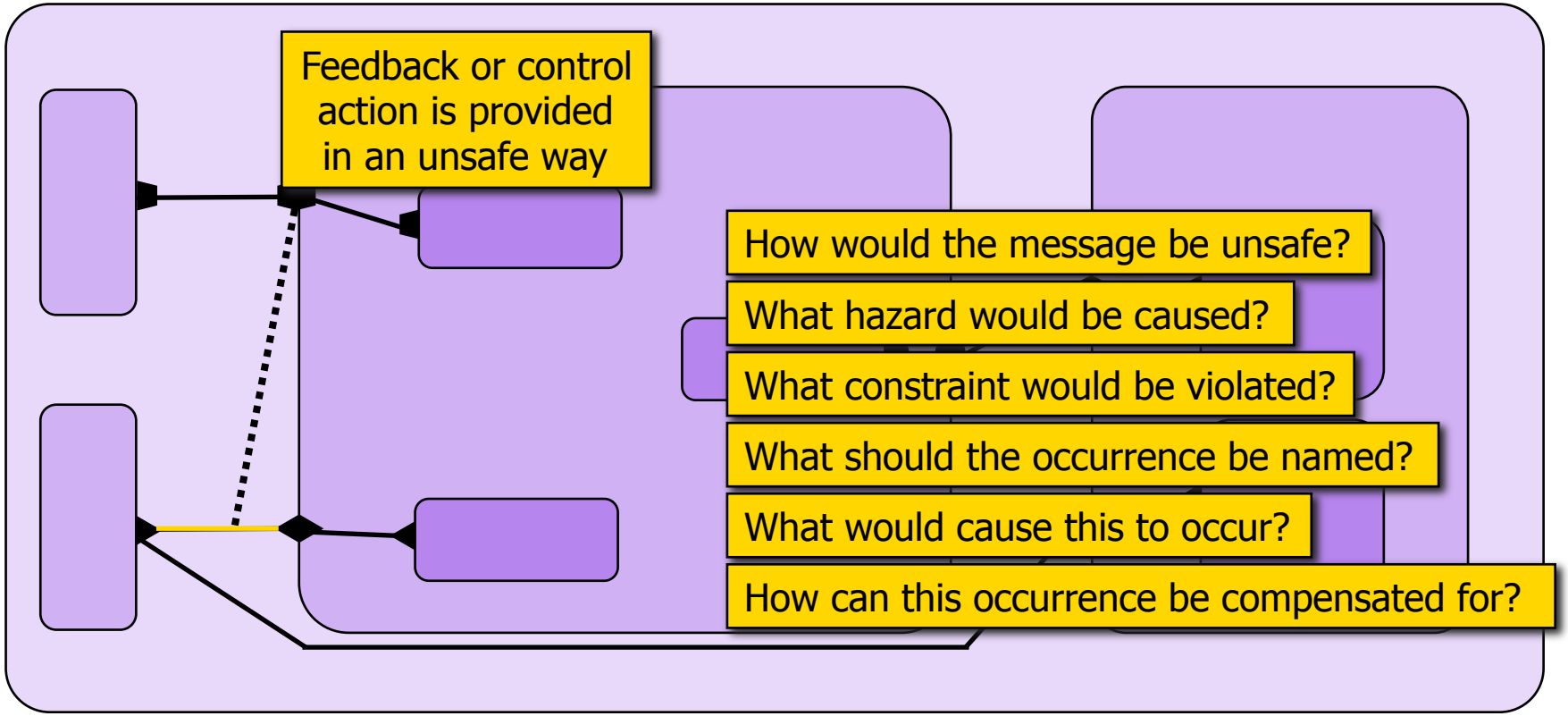
Step 2: Determining How Unsafe Control Actions Could Occur

Control Action: App -> Pump: Pump Normally

- Providing:
  - Bad Data:
    - Cause:
      - Incorrect values are gathered from one of the physiological sensors
    - Compensation:
      - Rely on multiple sensed physiological parameters to provide redundancy
- Not Providing:
  - Not hazardous

Hazard Analysis

Annotating our Architectural Model



Hazard Analysis

Annotating our Architectural Model

```
package PCA_Shutoff
public

system PCA_Shutoff_System
end PCA_Shutoff_System;

system implementation PCA_Shutoff_System.imp
subcomponents
   pulseOx : device PulseOx_Interface::ICEpoInterfa
   appLogic : process PCA_Shutoff_Logic::ICEpcaShut
connections
   spo2_data : port pulseOx.SpO2 -> appLogic.SpO2;
annex EMV2 {**
   use types PCA_Shutoff_Errors;
   properties
   MAP_Error_Properties::Occurrence => [
      Kind => AppliedTooLong;
      Hazard => PCA_Shutoff_Error_Properties::InadvertentPumpNormally;
      ViolatedConstraint => PCA_Shutoff_Error_Properties::PumpWhenSafe;
      Title => "Network Drop";
      Cause => "Network drops out, leaving the SpO2 value po
      Compensation => "Physiological readings have a maximum                    onger valid";
      Impact => reference(SpO2ValueHigh);
   ] applies to spo2_data;
**};

end PCA_Shutoff_System.imp;
end PCA_Shutoff;
```

How would the message be unsafe?

What hazard would be caused?

What constraint would be violated?

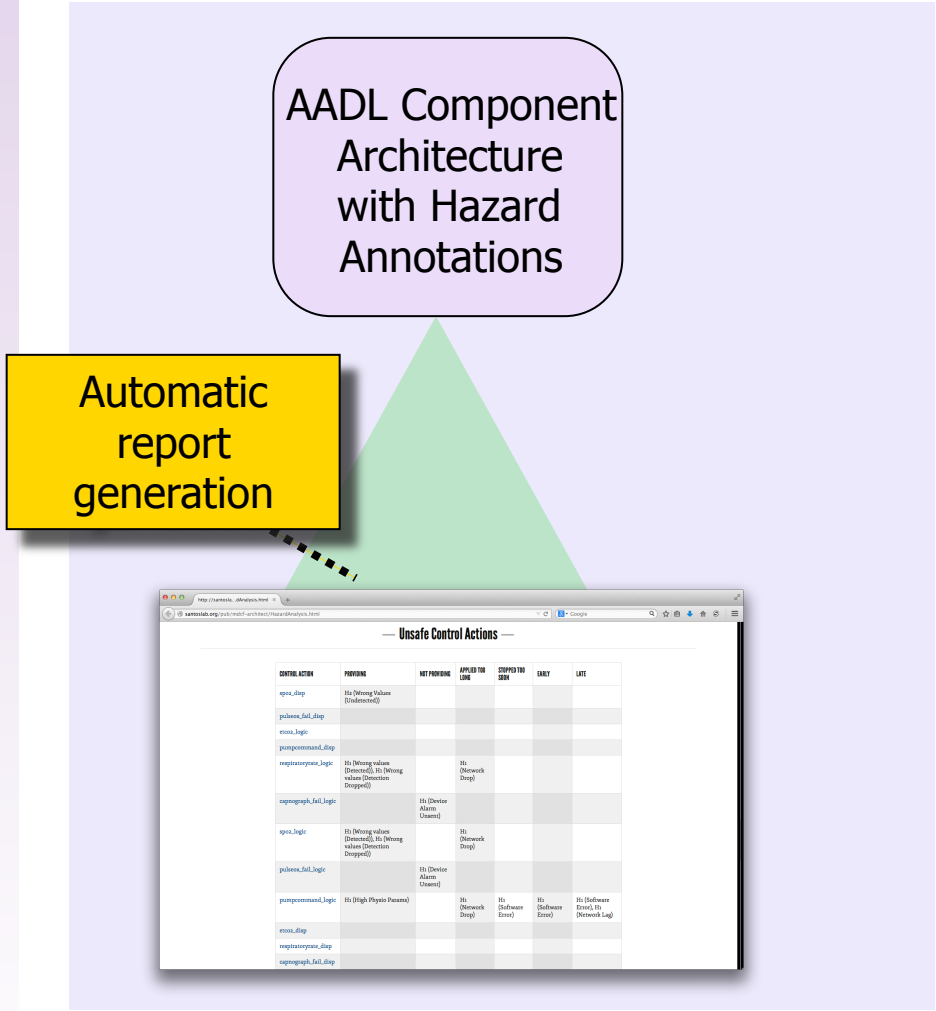
What should the occurrence be named?

What would cause this to occur?

How can this occurrence be compensated for?

We'll come back to these two in a moment.

Report Generation Development



- Development of component architecture using AADL / OSATE2
- Addition of Hazard Analysis Annotations
- Automatic generation of STPA-Styled Hazard Analysis Report

Example "In Progress" Report Online at:
http://santoslab.org/pub/mdcf-architect/HazardAnalysis.html

Annotating our Architectural Model

Inside the AADL System Component

```
package PCA_Shutoff
public

system PCA_Shutoff_System
end PCA_Shutoff_System;

system implementation PCA_Shutoff_System.imp
subcomponents
   pulseOx : device PulseOx_Interface::ICEpoInterface.imp;
   appLogic : process PCA_Shutoff_Logic::ICEpcaShutoffProcess.imp;
connections
   spo2_data : port pulseOx.SpO2 -> appLogic.SpO2;
annex EMV2 {**
   use types PCA_Shutoff_Errors;
   properties
   MAP_Error_Properties::Occurrence => [
      Kind => AppliedTooLong;
      Hazard => PCA_Shutoff_Error_Properties::InadvertentPumpNormally;
      ViolatedConstraint => PCA_Shutoff_Error_Properties::PumpWhenSafe;
      Title => "Network Drop";
      Cause => "Network drops out, leaving the SpO2 value potentially too high";
      Compensation => "Physiological readings have a maximum time, after which they are no longer valid";
      Impact => reference(SpO2ValueHigh);
   ] applies to spo2_data;
**};

end PCA_Shutoff_System.imp;
end PCA_Shutoff;
```

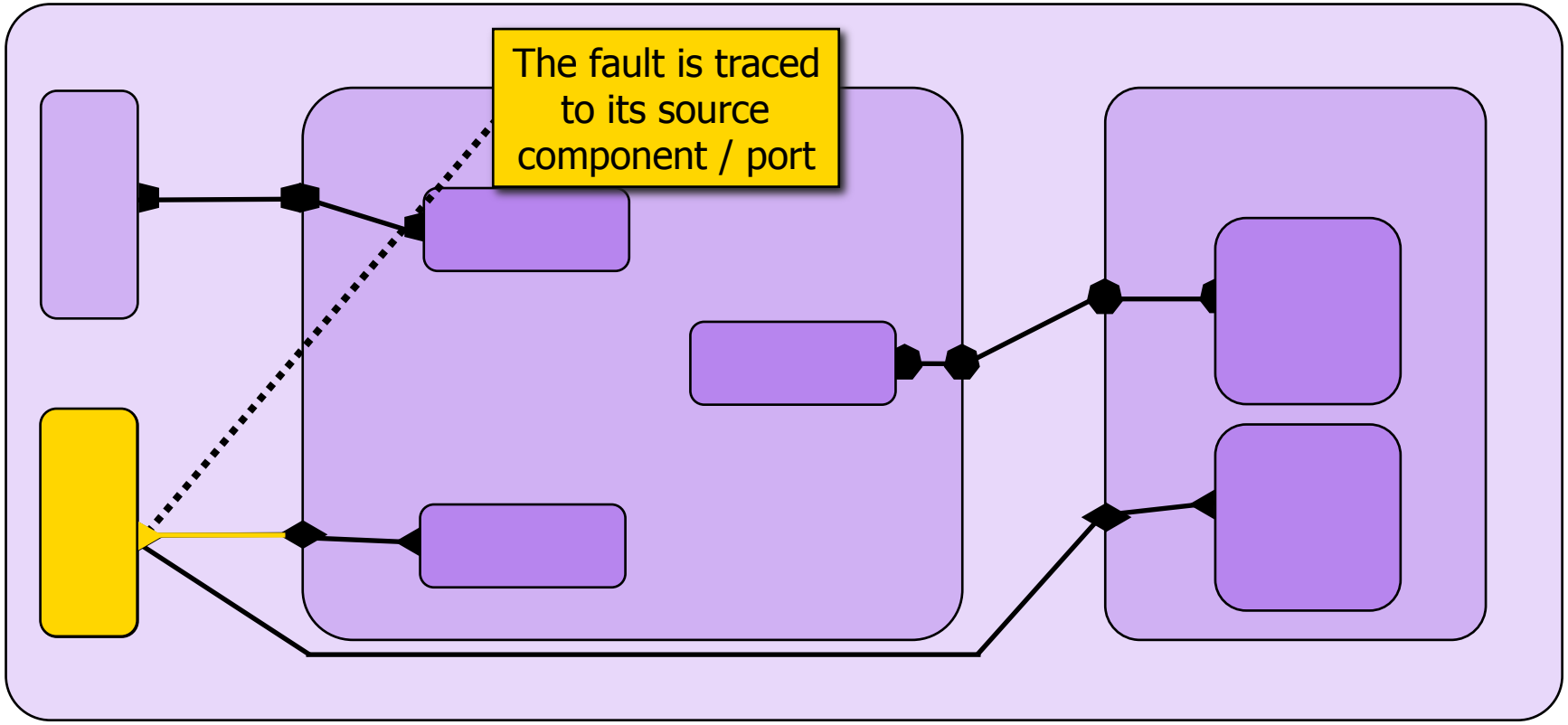
What channel will be affected?

What specific fault will result?

What can we do with our model + specific fault information?

Hazard Analysis

Annotating the Architectural Model



Hazard Analysis

Specification Step 1: Propagation

```
package PulseOx_Interface
public
with PCA_Shutoff_Types, PCA_Shutoff_Errors, EMV2, MAP_Error_Properties, PCA_Shutoff;
   device ICEpoInterface
   features
      SpO2 : out event data port PCA_Shutoff_Types::SpO2;
   annex EMV2 {**
      use types PCA_Shutoff_Errors;
      error propagations
         SpO2 : out propagation {SpO2ValueHigh};
         flows
            SpO2UnDetectableHighValueFlowSource : error source SpO2 {SpO2ValueHigh};
      end propagations;
   **};
   end ICEpoInterface;

   device implementation ICEpoInterface.imp
   end ICEpoInterface.imp;

end PulseOx_Interface;
```

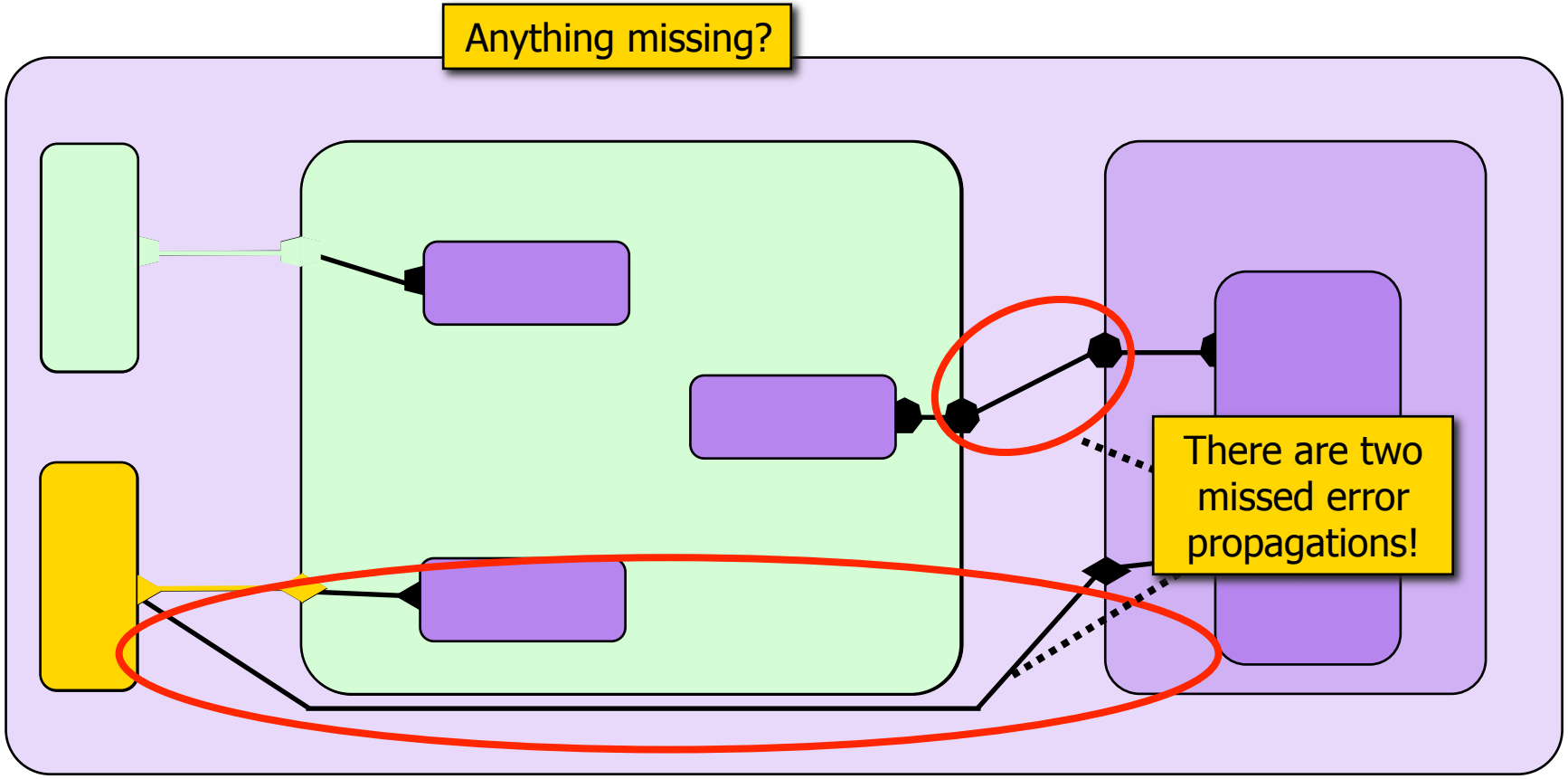
Port the fault will propagate on

Specific Fault

Direction of the propagation

Hazard Analysis

Annotating the Architectural Model



Hazard Analysis

OSATE Remembers A Neglected Connection

```
system implementation PCA_Shutoff_System.imp
subcomponents
    -- Physiological inputs
    pulseOx : device PulseOx_Interface::ICEpoInterface.imp;

    -- App logic
    appLogic : process PCA_Shutoff_Logic::ICEpcaShutoffProcess.imp;
    appDisplay : process PCA_Shutoff_Display::ICEpcaDisplayProcess.imp;
connections
    -- From components to logic
    spo2_logic : port pulseOx.SpO2 -> appLogic.SpO2;

    -- From components to display
    spo2_disp : port pulseOx.SpO2 -> appDisplay.SpO2;
anne
```

No incoming error propagation from appDisplay for outgoing propagation SpO2{SpO2ValueHigh}. Check for Unhandled Faults.

```
    properties
    -- Errors between the PulseOx's SpO2 channel and the App Logic
    MAP_Error_Properties::Occurrence => [
        Kind => ValueHigh;
        Hazard => PCA_Shutoff_Error_Properties::PatientHarmed;
        ViolatedConstraint => PCA_Shutoff_Error_Properties::PumpWhenSafe;
        Title => "Wrong Values (Undetected)";
        Cause => "Incorrect values are gathered from the physiological sensors";
```

Hazard Analysis

Interaction between Report and Model

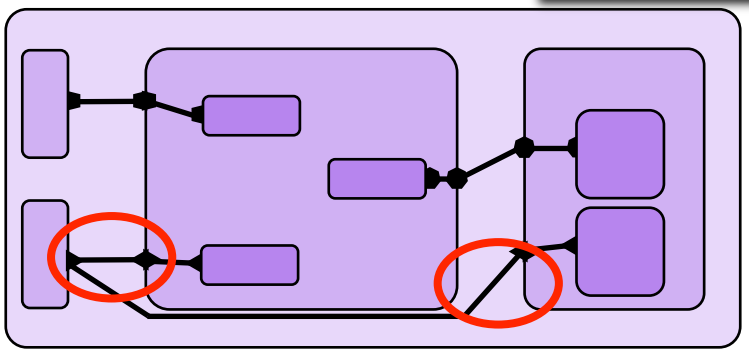
CONTROL ACTION	PROVIDING	NOT PROVIDING	APPLIED TOO LONG	STOPPED TOO SOON	EARLY	LATE
spo2_disp	H2 (Wrong Values (Undetected))					
pulseox_fail_disp						
etco2_logic						
pumpcommand_disp						
respiratoryrate_logic	H1 (Wrong values (Detected)), H1 (Wrong values (Detection Dropped))		H1 (Network Drop)			
capnograph_fail_logic		H2 (Alarm Unsent)				
spo2_logic	H1 (Wrong values (Detected)), H1 (Wrong values (Detection Dropped))		H1 (Network Drop)			

Bottom Up

Top Down

1. Report indicates analysis incomplete

2. Developer creates occurrence property and supporting EMV2 annotations



3. Tool highlights unconsidered propagation paths

4. Developer creates supporting occurrence property, considers alternative impacts of hazard

Impacts

- Automation
  - Traditionally, analysts have to mine a system and maintain it – without tool support
- Architectural integration
  - Faults can be “bound” to specific components and ports
- Future:
  - Testing + Fault Injection
    - If a compensation is claimed, we can auto-generate a test

An Architecturally-Integrated, Systems-Based Hazard Analysis for Medical Applications

http://cis.ksu.edu/~samprocter

Sam Procter and John Hatcliff
SAnToS Lab
Kansas State University

Support:

This work is supported in part by the US National Science Foundation (NSF) (#1239543), the NSF US Food and Drug Administration Scholar-in-Residence Program (#1355778) and the National Institutes of Health / NIBIB Quantum Program.