Using STPA to Support Risk Management for Interoperable Medical Systems

STAMP Workshop 2015, MIT

Sam Procter, John Hatcliff
SAnToS Lab
Kansas State University

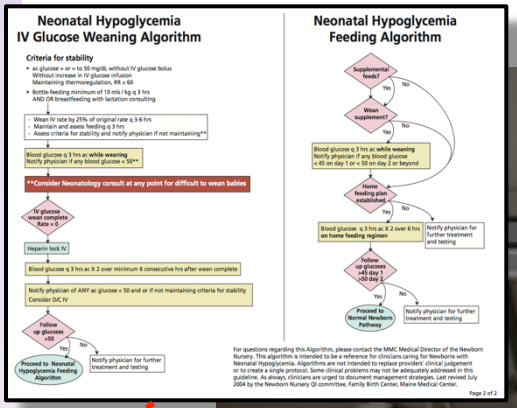
Anura Fernando
Underwriters Laboratories

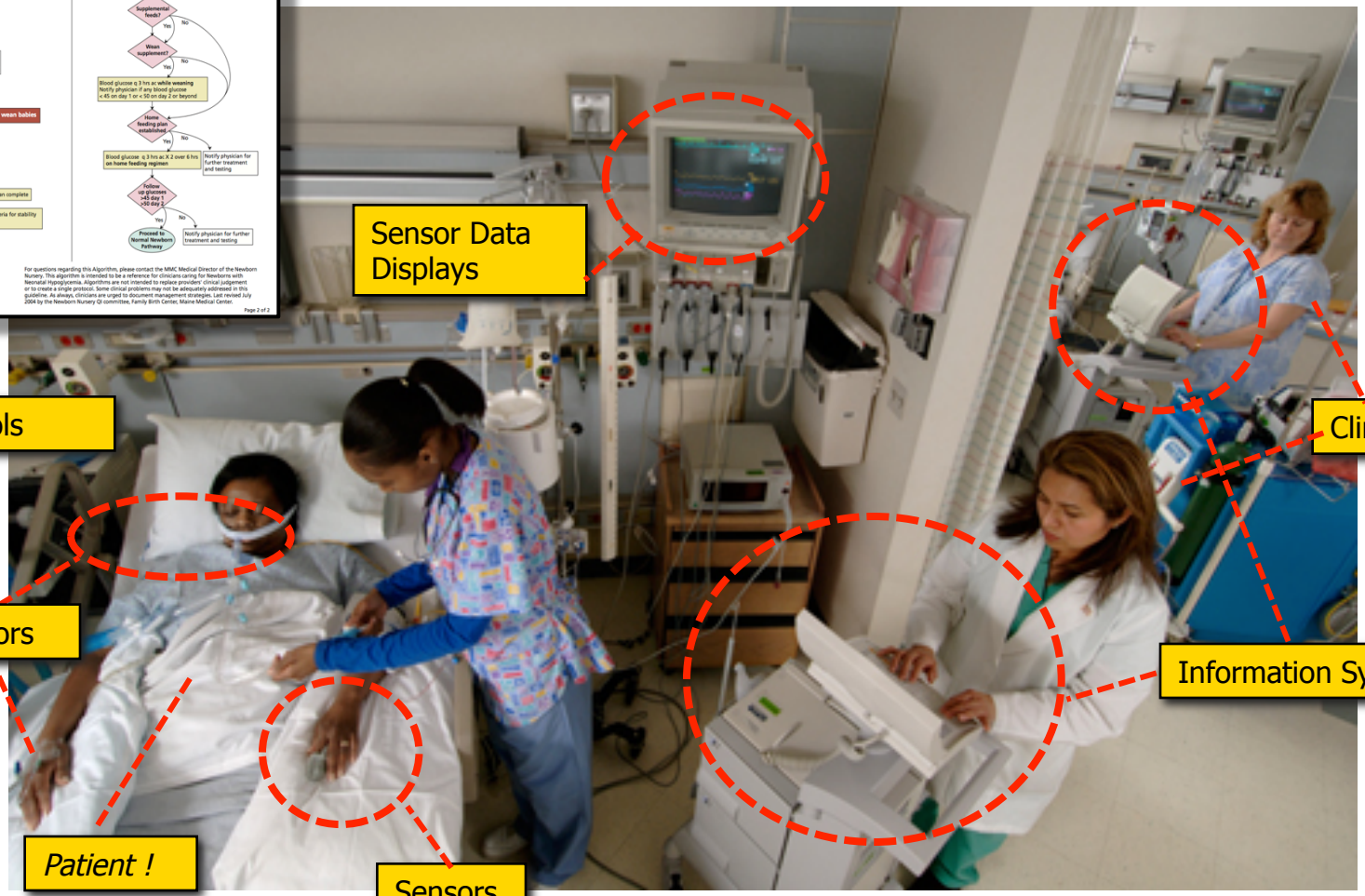
Sandy Weininger
US Food and Drug Administration

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

Motivation

- What are the types of things we could do with device integration?
  - Information forwarding
  - Automation of clinical workflows
  - Closed loop control between devices
- Unlike personal computing, medical devices are not designed to work together
- Integrating medical devices would bring myriad benefits

- … how can we do so safely?

Outline

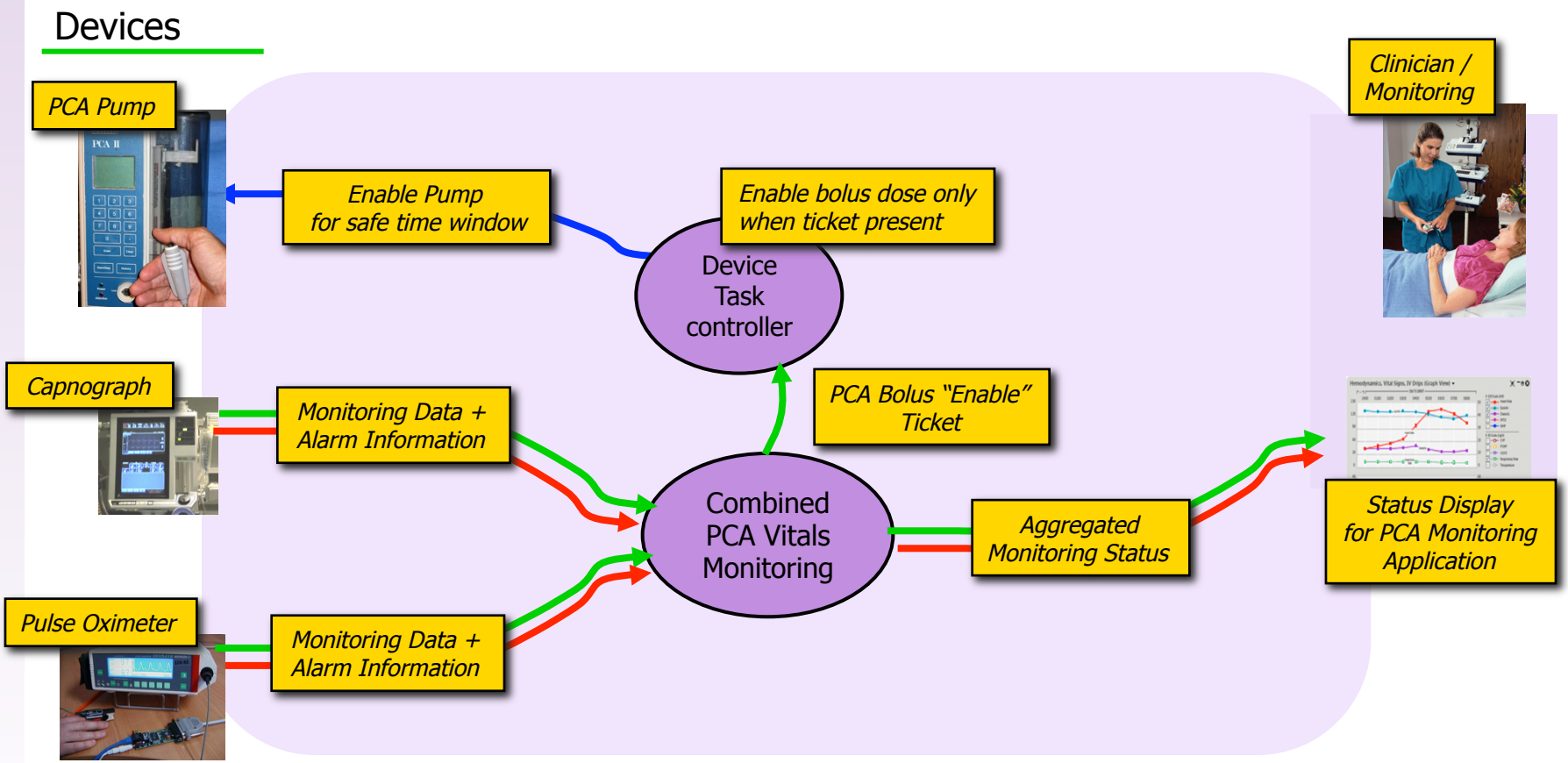
- Background
  - PCA Interlock Scenario
  - Medical Application Platforms
  - Tooling
- Hazard Analysis In AADL
- Architectural Integration

PCA Interlock Scenario

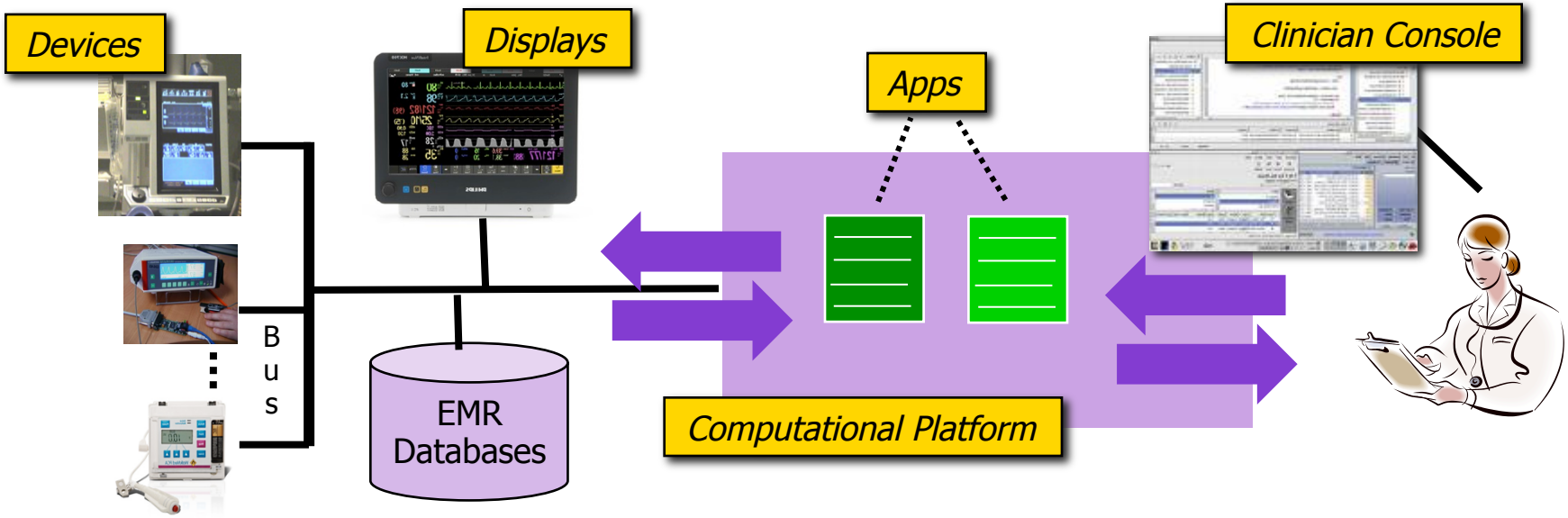
- Patients are commonly given patient-controlled analgesics after surgery
- Crucial to care, but numerous issues related to safety
- Data for disabling the pump exists now (just a system invariant) -- we just need to integrate it

PCA Pump Safety Interlock

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



Medical Application Platforms



- A *Medical Application Platform* is a safety- and security-critical real-time computing platform for…
  - Integrating heterogeneous devices, medical IT systems, and information displays via communications infrastructure, and
  - Hosting applications (“apps”) that provide medical utility via the ability to acquire information from and update/control integrated devices, IT systems, and displays

Unique aspects of MAP domain

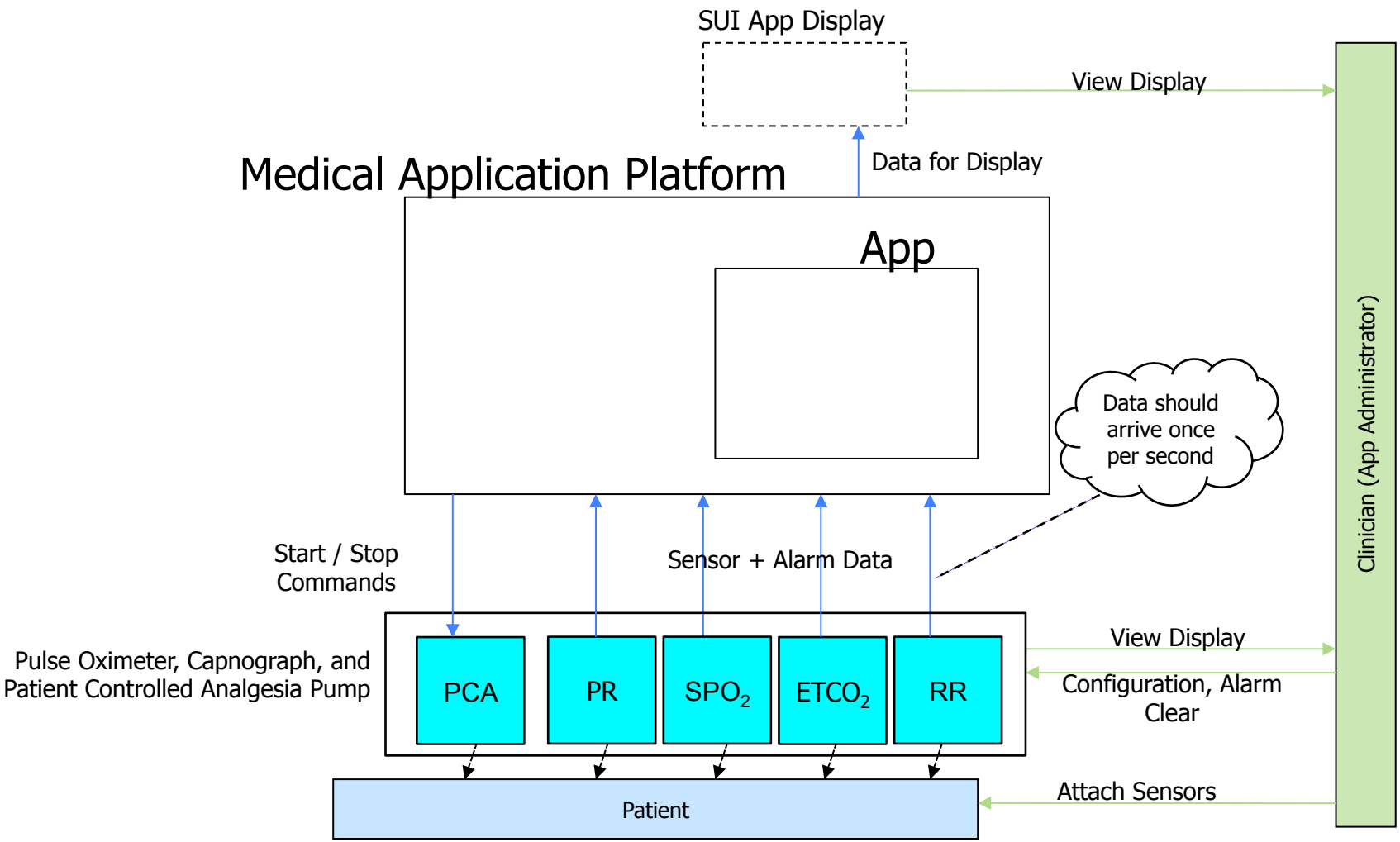
- Software based
  - Hardware is interchangeable
- Component oriented
- Unclear how FTA / FMEA might apply
- Early, firm notion of system architecture
  - Standardized in UL 2800

Extension beyond medicine

- We use medicine in our examples
  - … but this can extend to other compositional systems
- Core idea:
  - Integration of heterogeneous
    - Sensors,
    - Actuators, and
    - Complete systems,
  - by small chunks of software,
  - in a verifiable manner

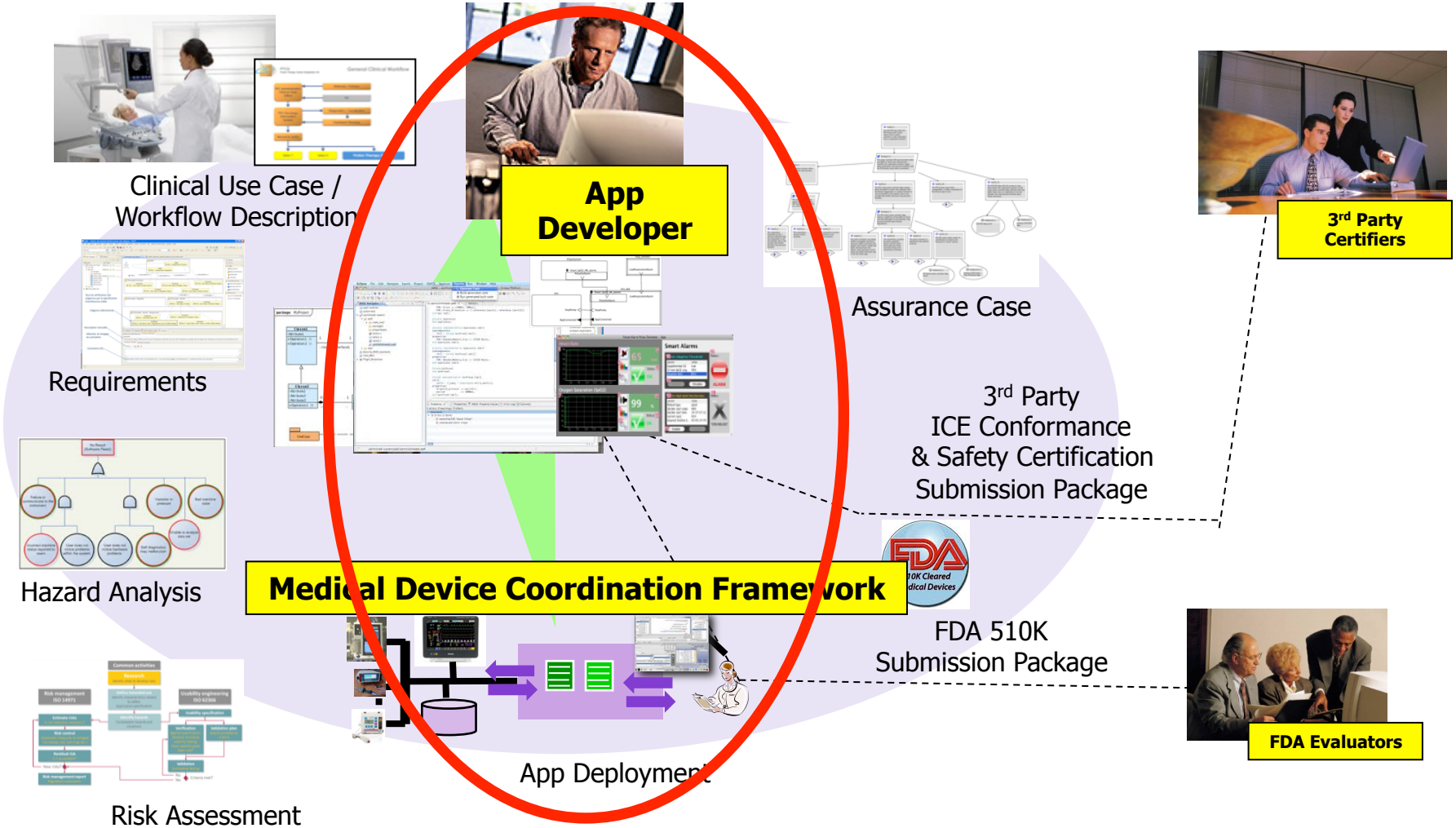
Background

PCA Pump Interlock Architecture



Tooling Vision

Analyses and Regulatory Artifacts



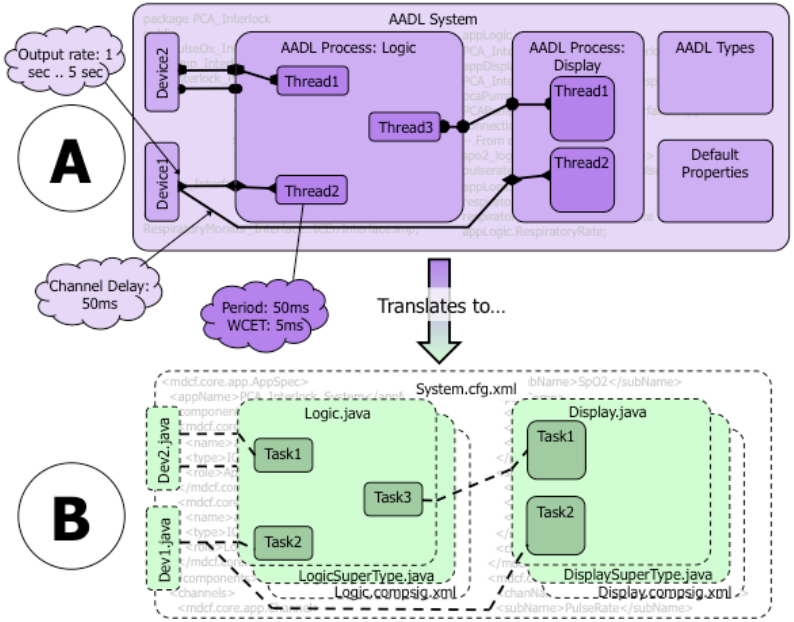
Tooling Vision

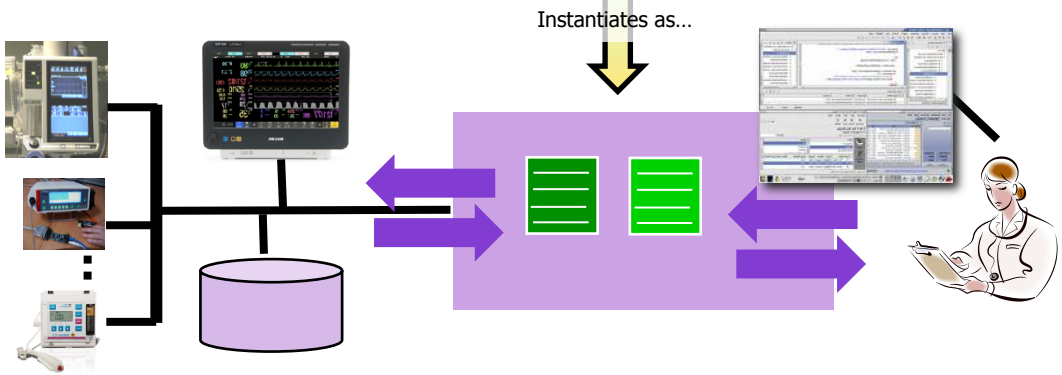
Code Generation

A. The app's architecture is specified in a suitable formalism
   1. Components as AADL Devices / Processes
   2. Connections are specified
   3. RT/QoS Parameters are via AADL's property-specification mechanism

B. The app is programmatically translated to Java and XML
   1. Only "Business Logic" is written by the developer

C. The app is launched on a compatible MAP



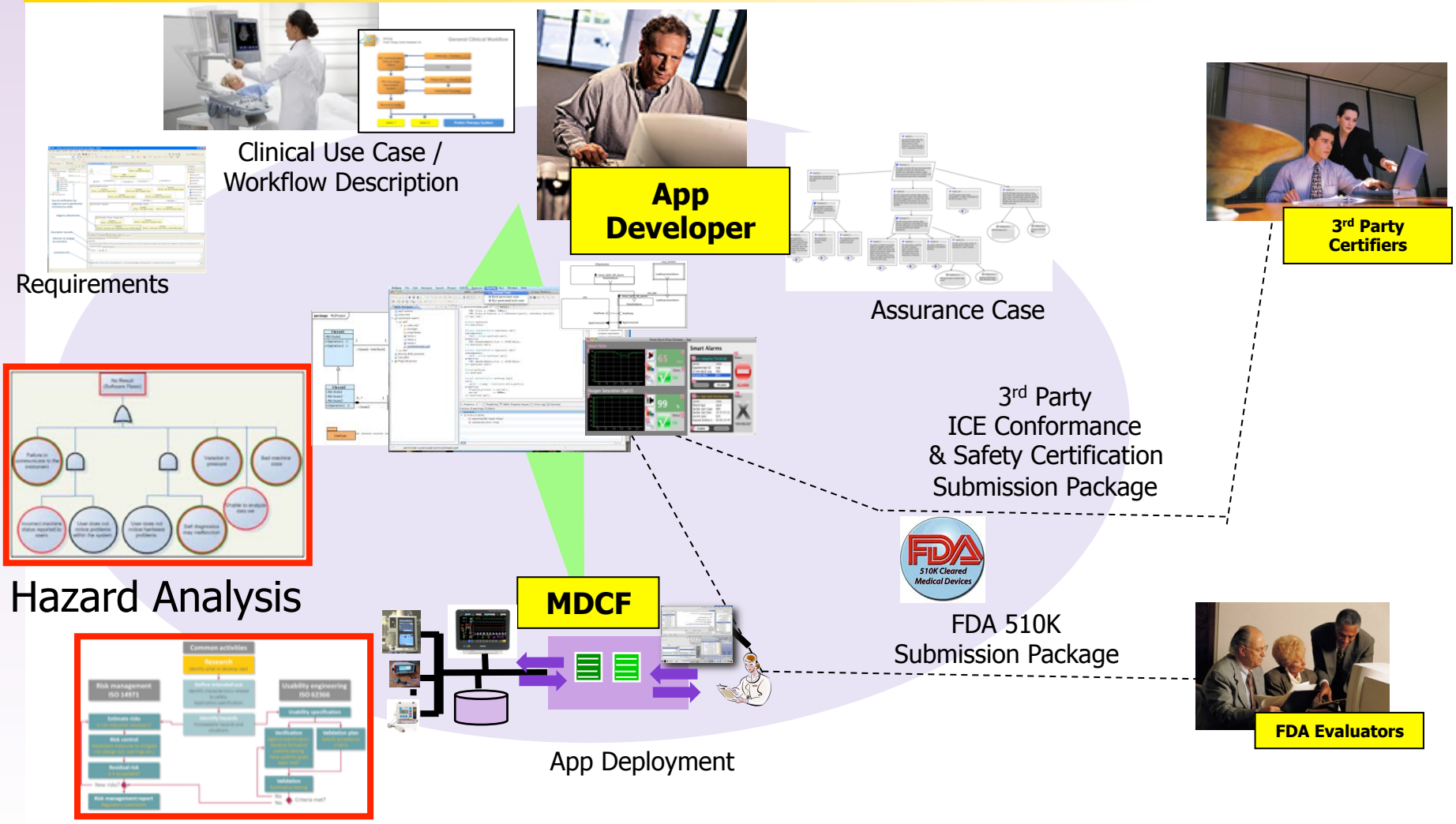


Outline

- Background
- Hazard Analysis In AADL
  - Correspondence with manual HA
  - STPA Fundamentals
  - Report Generation
- Architectural Integration

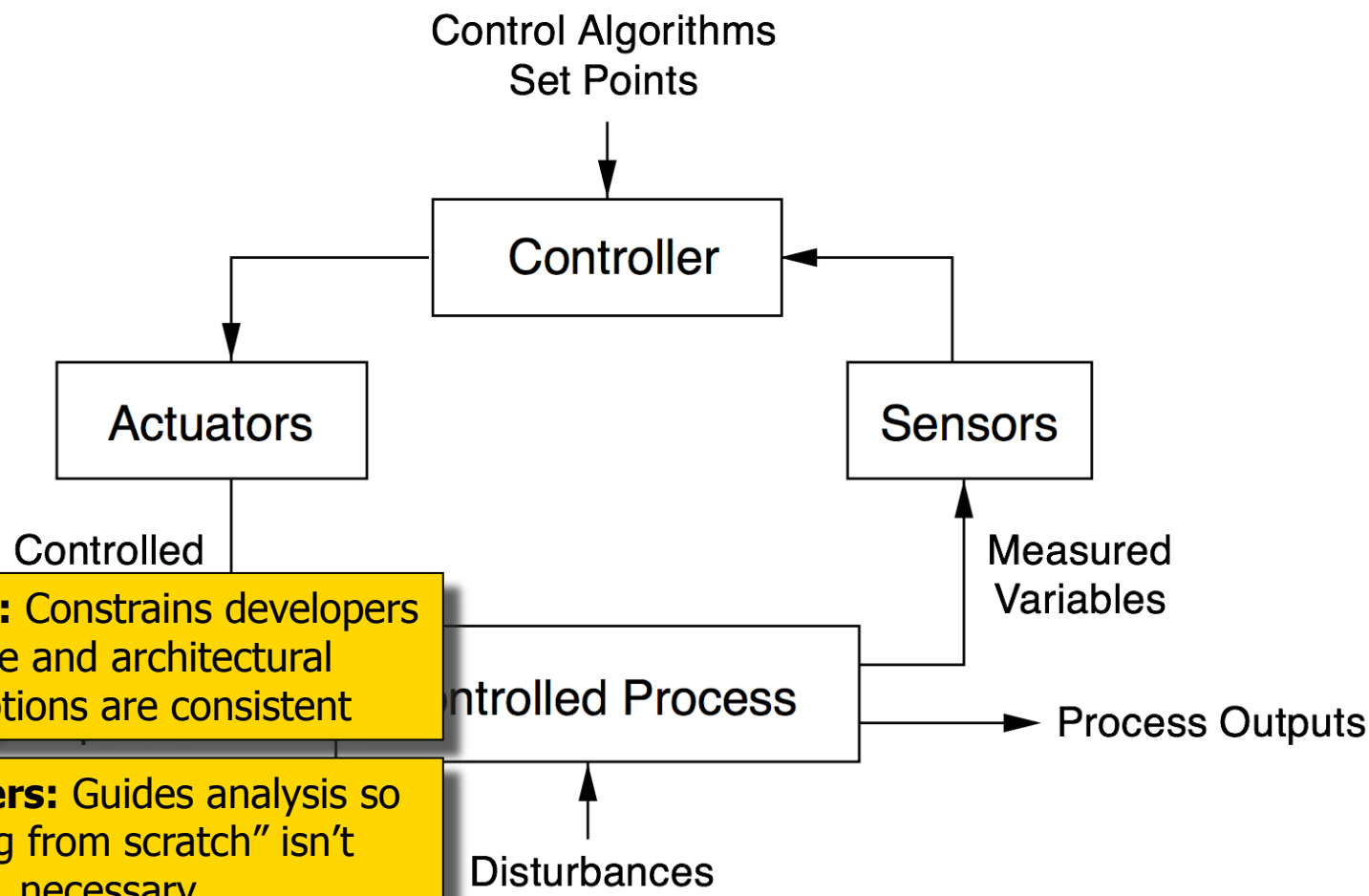
Hazard Analysis

Leveraging Semiformal Architectural Descriptions



Hazard Analysis in AADL

What if we could draw control loops with code?



Benefits:

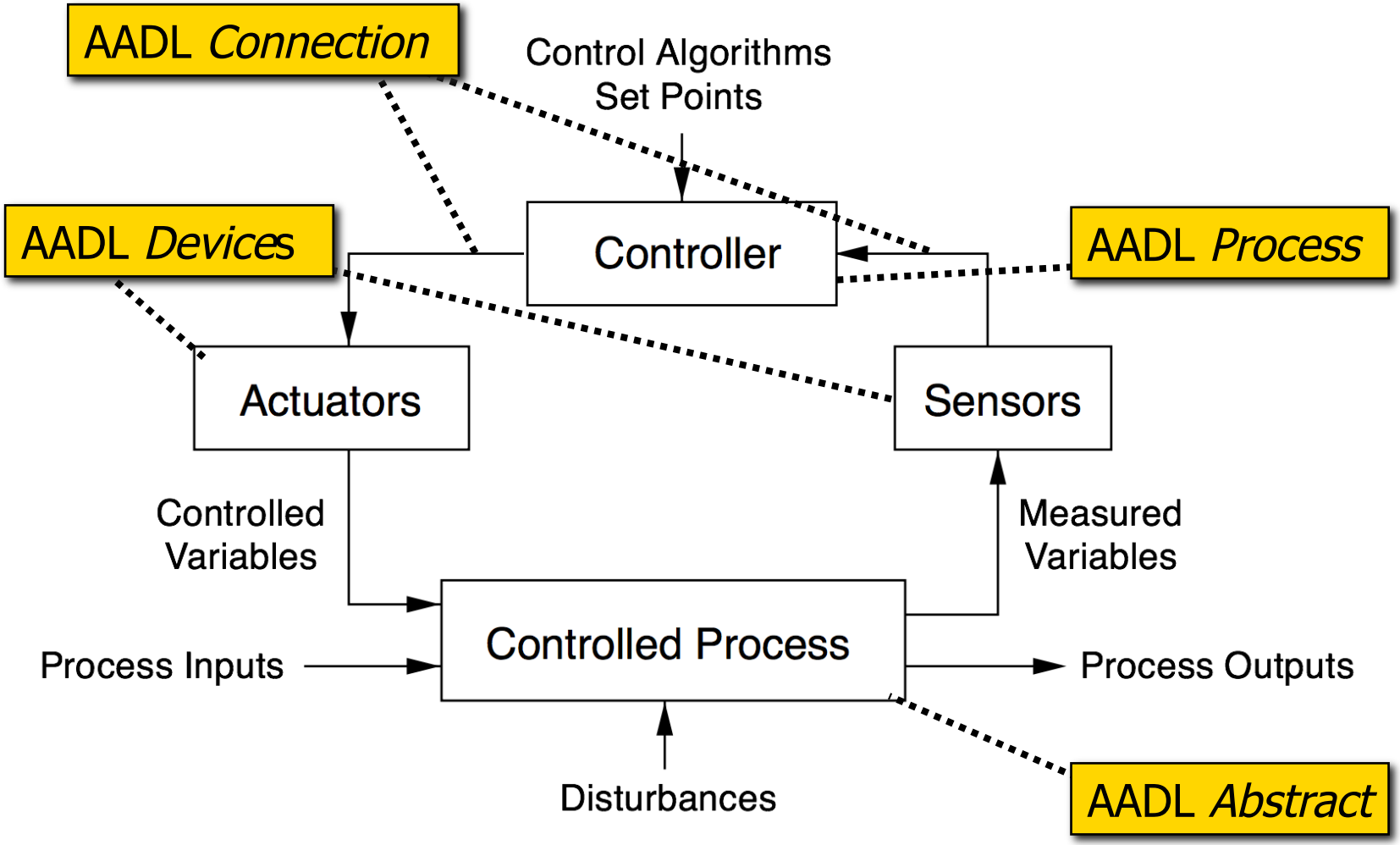
Managers: Constrains developers so style and architectural assumptions are consistent

Developers: Guides analysis so "starting from scratch" isn't necessary

Nancy Leveson. Figure 3.2, Page 66, *Engineering A Safer World*. MIT Press, 2011

Hazard Analysis in AADL

AADL Equivalents of STPA's Objects



Nancy Leveson. Figure 3.2, Page 66, *Engineering A Safer World*. MIT Press, 2011

STPA in AADL

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 [PCADoS]

```
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;
];
```

Benefits:

Regulators: Supports strong traceability both in code and in (hypertext) reports

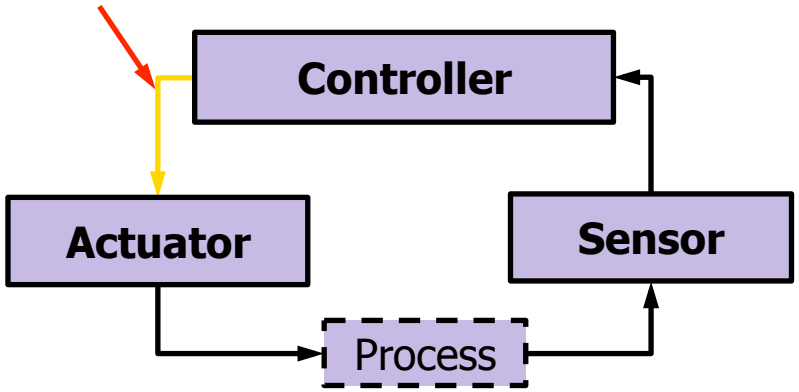
STPA in AADL

Fundamentals

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

Example

- App -> Pump: Pump Normally



Benefits:

Developers: Hazard Analysis artifacts are automatically in-sync with system architecture

STPA in AADL

Identifying Hazardous Control Actions

- Hazardous Control Action Table
  - 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

PH = Patient Harmed
BID = Bad Info Displayed

STPA in AADL

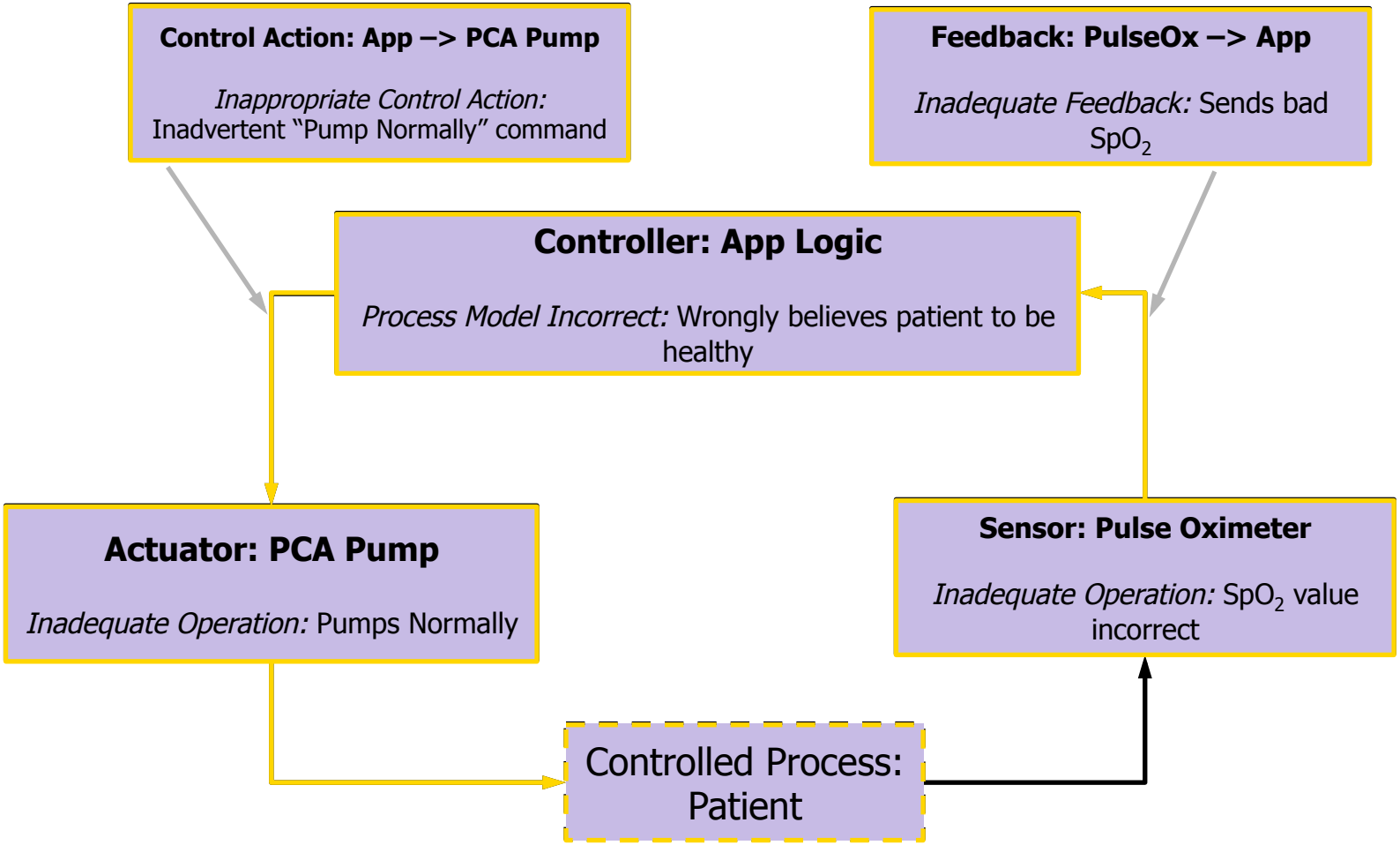
Hazardous Causes and Compensations

Control Action: App -> Pump: Pump Normally

- Providing:
  - Inadequate Sensor Operation:
    - 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

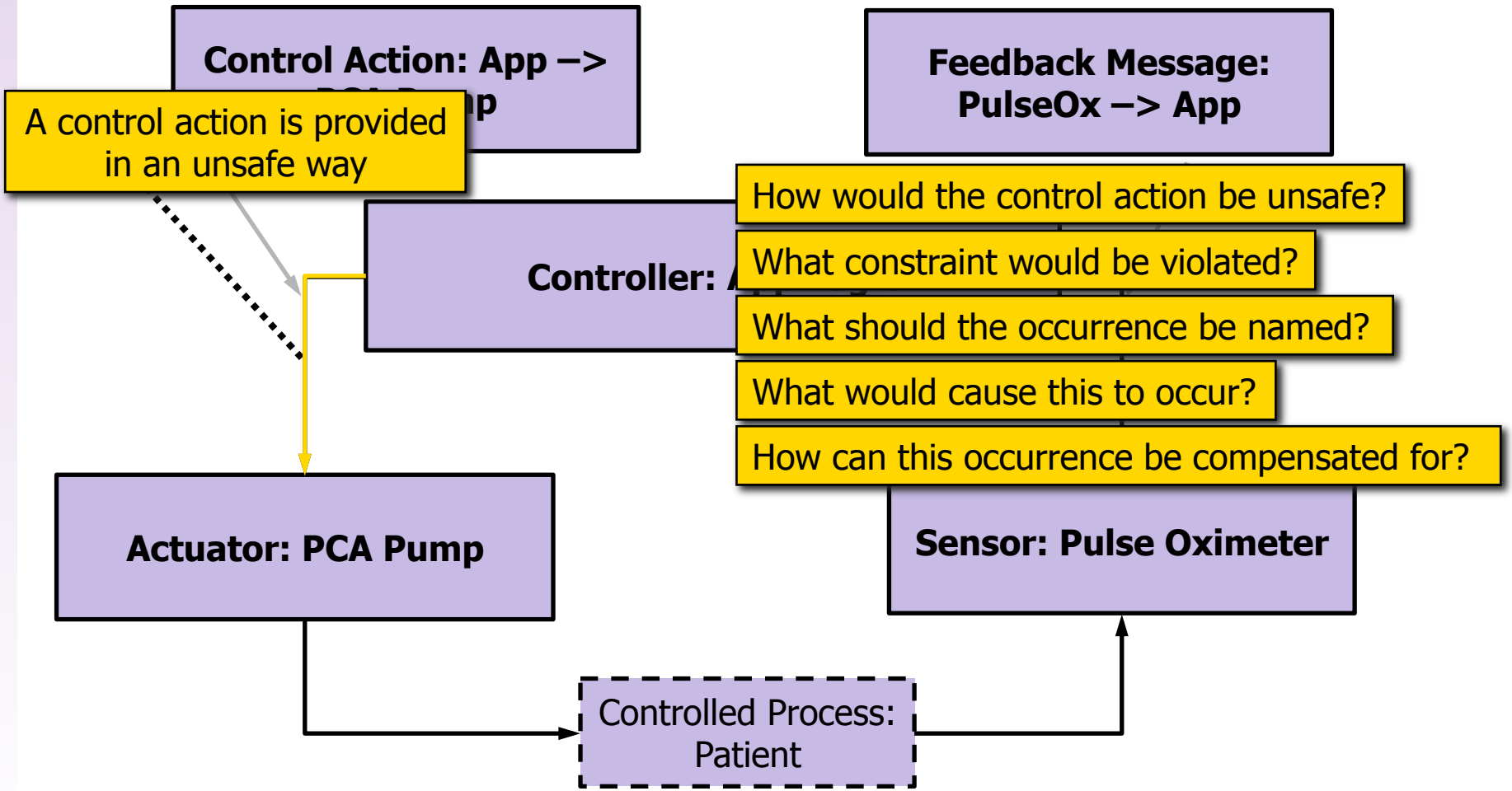
STPA in AADL

The Annotated Control Loop



STPA in AADL

Where should we start?



Hazard Analysis

Annotating our Architectural Model

```
package PCA_Interlock_System
public

system PCA_Interlock_System
end PCA_Interlock_System;

system implementation PCA_Interlock_System.imp
subcomponents
  pulseOx : device PulseOx_Interface::MAP_PulseOx_
  pcaPump : device PCAPump_Interface::MAP_PCAPump_
  appLogic : process PCA_Interlock_Logic::PCA_Inte
connections
  spo2_data : port pulseOx.SpO2 -> appLogic.SpO2;
  pump_cmd : port appLogic.pumpCmd -> pcaPump.cmd;
annex EMV2 {**
  use types PCA_Interlock_Errors;
  properties
  MAP_Error_Properties::Occurrence => [
    Guideword => Providing;
    ViolatedConstraint => PCA_Shutoff_Error_Properties::DontLe
    Title => "High Physio Params";
    ErrorType => reference(InadvertentPumpNormally);
    Description => "One or more physiological parameters are t
incorrectly believe the patient is healthy";
    Compensation => "Physiological values are cross-checked with others";
  ] applies to pump_cmd;
**};

end PCA_Interlock_System.imp;
end PCA_Interlock_System;
```

How would the control action be unsafe?

What constraint would be violated?

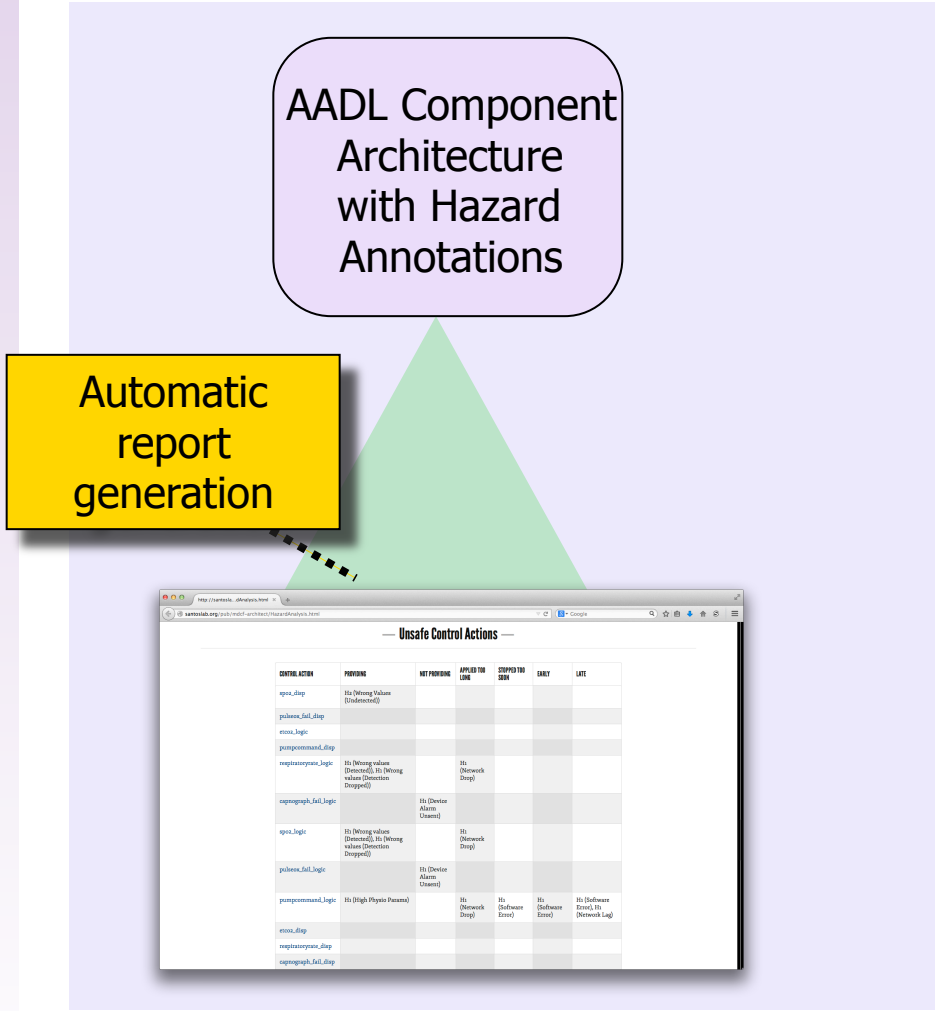
What should the occurrence be named?

What would cause this to occur?

How can this occurrence be compensated for?

We'll come back to this one 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
- Very strong traceability between system and HA report

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

Automatic Report Generation

Fundamentals

Accident Levels

1. **AL1**: Death or serious injury to a human

Accidents

1. **A1**: Patient is killed or seriously injured. [AL1]

Hazards

1. **H1**: Commands for dosage exceeding the patient's tolerance are sent to the pump. [A1]
2. **H2**: Incorrect information is sent to the display. [A1]

Safety Constraints

1. **C3**: The app must inform the display of the pump command status. [H2]
2. **C1**: The app must command the pump to stop if the patient's vital signs indicate over-infusion. [H1]
3. **C2**: The app must inform the display of the status of the patient's vital signs. [H2]

Automatic Report Generation

Unsafe Control Action Table

— Unsafe Control Actions —

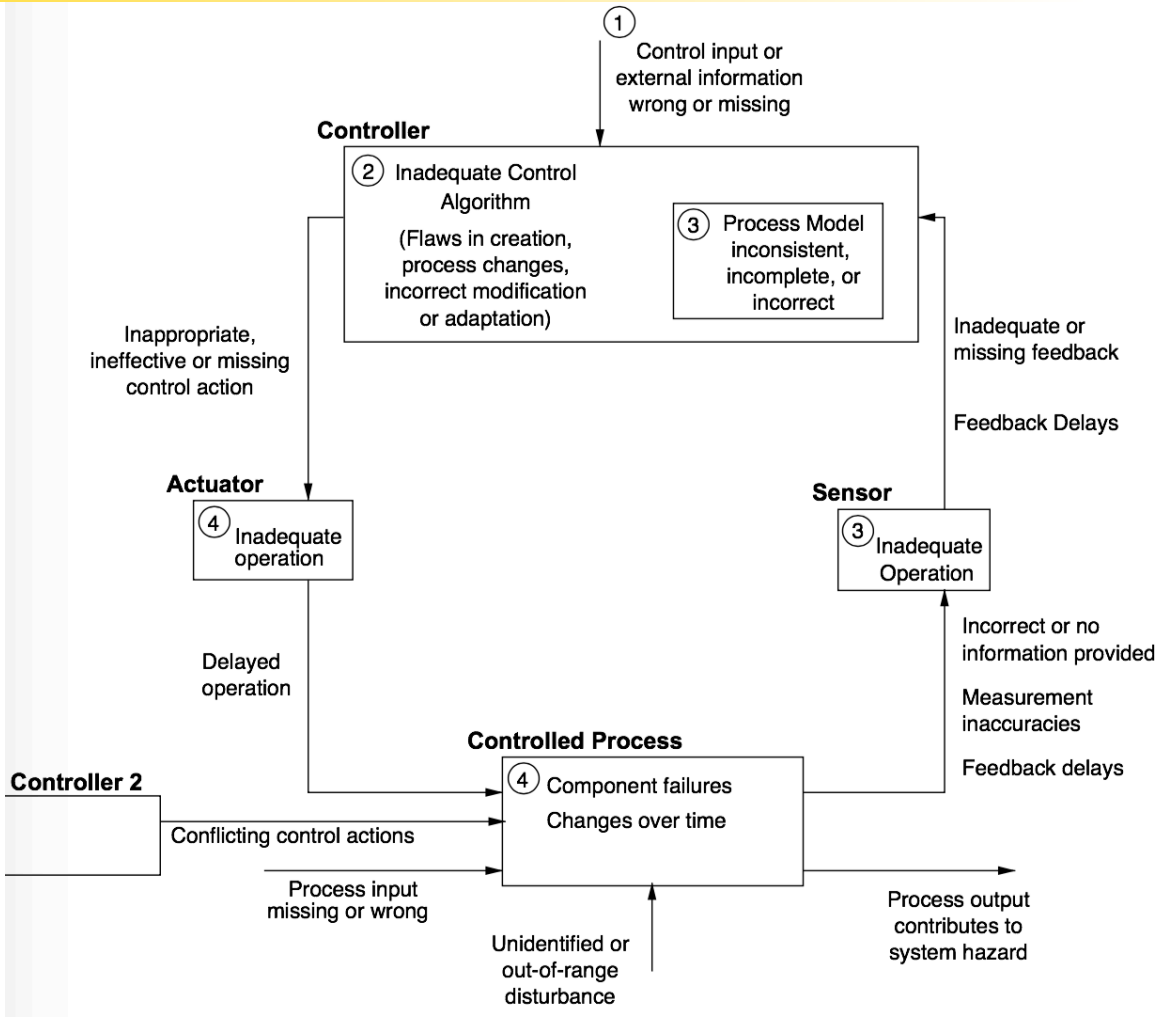
CONTROL ACTION	PROVIDING	NOT PROVIDING	APPLIED TOO LONG	STOPPED TOO SOON	EARLY	LATE
spo2_disp	H2 (Wrong Values (Undetected))					
pulseox_fail_disp						
etco2_logic						
pumpcommand_disp						
respiratoryrate_logic	H1 (Wrong values (Detected)), H1 (Wrong values (Detection Dropped))		H1 (Network Drop)			
capnograph_fail_logic		H1 (Device Alarm Unsent)				
spo2_logic	H1 (Wrong values (Detected)), H1 (Wrong values (Detection Dropped))		H1 (Network Drop)			
pulseox_fail_logic		H1 (Device Alarm Unsent)				
pumpcommand_logic	H1 (High Physio Params)		H1 (Network Drop)	H1 (Software Error)	H1 (Software Error)	H1 (Software Error), H1 (Network Lag)
etco2_disp						
respiratoryrate_disp						
capnograph_fail_disp						

Outline

- Background
- Hazard Analysis In AADL
- Architectural Integration
  - EM Fault Types
  - Deeply Integrated Hazard Analysis
  - Tool Support

STPA's Causality Guidewords

Annotated Control Loop



Nancy Leveson. Figure 4.8, Page 93, *Engineering A Safer World*. MIT Press, 2011

AADL EM Fault Types

Type Hierarchy

Error Library Type	STPA Error Type	App Error Type
Errors with Physiological Monitors		
LateDelivery	DelayedOperation	SpO2ValueLate
IncorrectValue	IncorrectInformation	SpO2ValueLow
N/A	NoInformation	NoSpO2Data
Errors with App Logic		
ServiceCommission	InnapropriateCtrlAction	InadvertentPumpNormally
ServiceOmission	MissingCtrlAction	InadvertentPumpMinimally
AADL Standard Error Types	STPA Guidewords	App Specific Error Types

AADL EM Fault Types

App Specific Error Library

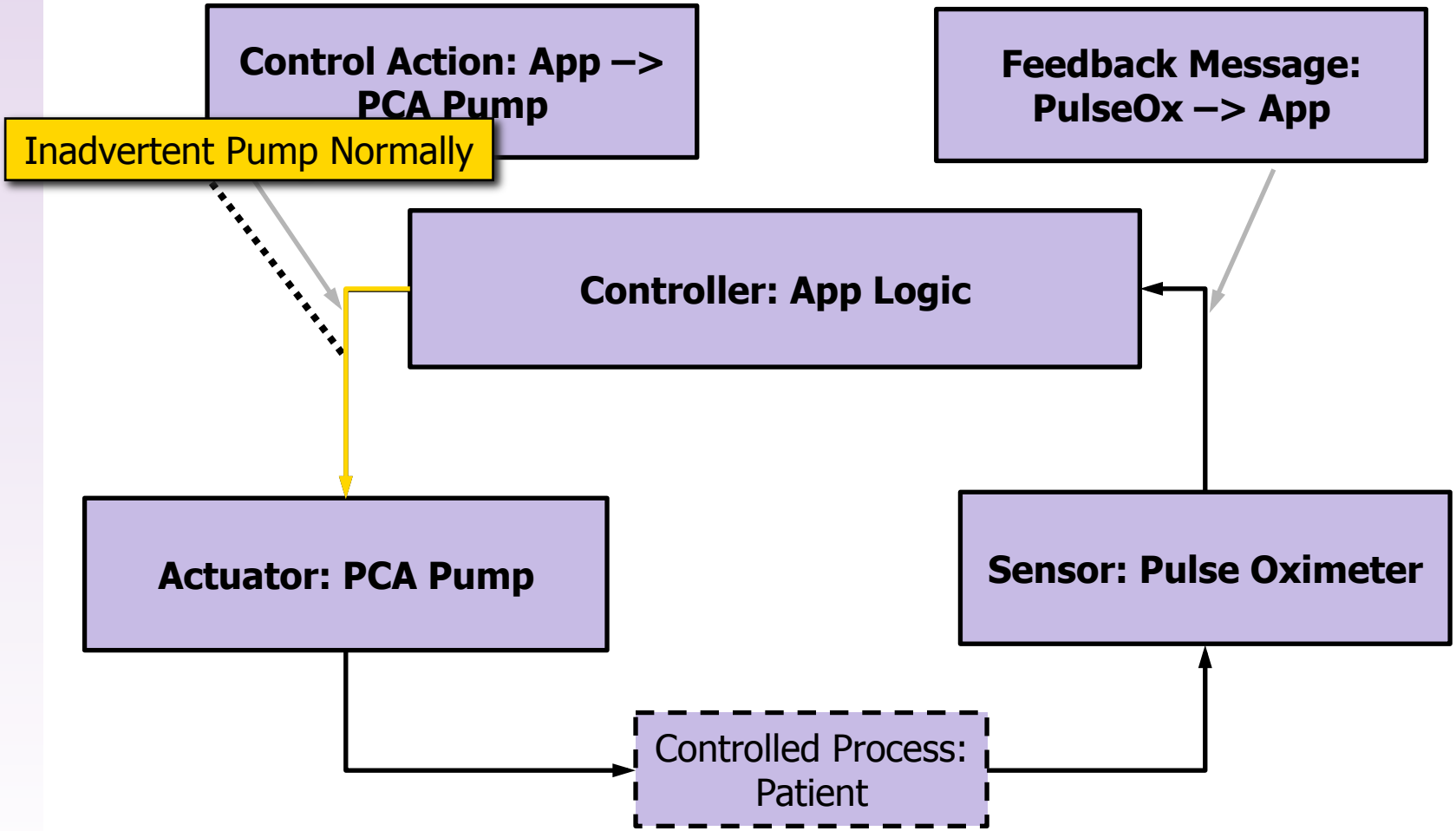
```
package PCA_Shutoff_Errors
public
with MAP_Errors, PCA_Shutoff_Error_Properties, MAP_Erro
        PCA_Shutoff;

annex EMV2
{**
        error types
                InadvertentPumpNormally : type extends MAP_Errors::InappropriateControlAction;

                -- Could also be inadequate feedback
                SpO2ValueHigh : type extends MAP_Errors::InadequateSensorOperation;
                SpO2ValueLow : type extends MAP_Errors::InadequateSensorOperation;
                ETCO2ValueLow : type extends MAP_Errors::InadequateSensorOperation;
                ETCO2ValueHigh : type extends MAP_Errors::InadequateSensorOperation;
                RespiratoryRateLow : type extends MAP_Errors::InadequateSensorOperation;
                RespiratoryRateHigh : type extends MAP_Errors::InadequateSensorOperation;
                DeviceAlarmFailsOn : type extends MAP_Errors::InadequateSensorOperation;
                DeviceAlarmFailsOff : type extends MAP_Errors::InadequateSensorOperation;
        end types;
**};

end PCA_Shutoff_Errors;
```

Application independent: Sourced from STPA

Application specific: Defined by app risk management process

STPA in AADL

Using our fault type



Integrated Hazard Analysis

Using our fault type

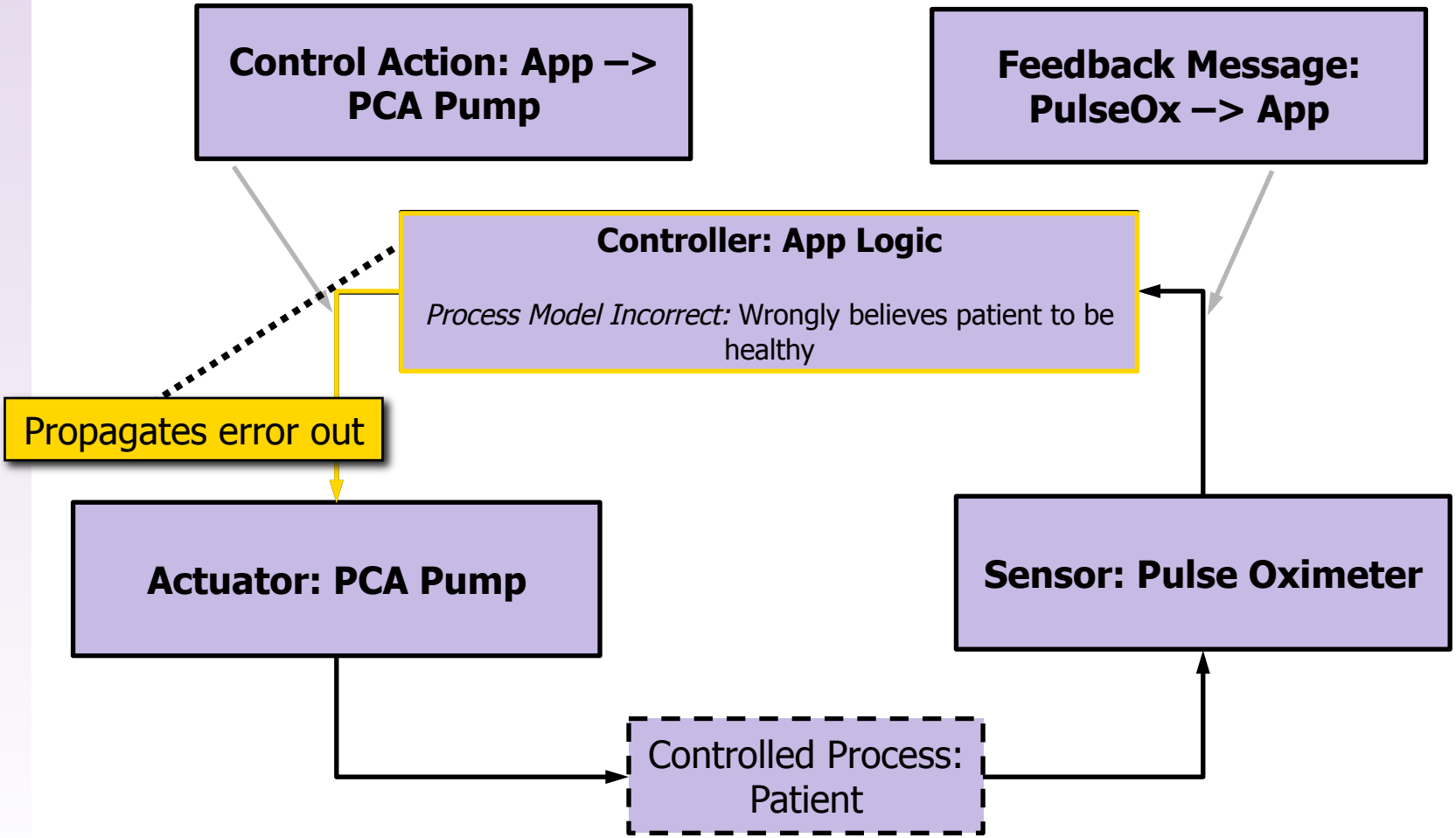
```
package PCA_Interlock_System
public

system PCA_Interlock_System
end PCA_Interlock_System;

system implementation PCA_Interlock_System.imp
subcomponents
  pulseOx : device PulseOx_Interface::MAP_PulseOx_Interface.imp;
  pcaPump : device PCAPump_Interface::MAP_PCAPump_Interface.imp;
  appLogic : process PCA_Interlock_Logic::PCA_Interlock_Logic.imp;
connections
  spo2_data : port pulseOx.SpO2 -> appLogic.SpO2;
  pump_cmd : port appLogic.pumpCmd -> pcaPump.cmd;
annex EMV2 {**
  use types PCA_Interlock_Errors;
  properties
  MAP_Error_Properties::Occurrence => [
    Guideword => Providing;
    ViolatedConstraint => PCA_Shutoff_Error_Properties::DontLetPumpRunWhenUnsafe;
    Title => "High Physio Params";
    ErrorType => reference(InadvertentPumpNormally);
    Description => "One or more physiological parameters are too high, leading the app logic to
incorrectly believe the patient is healthy";
    Compensation => "Physiological values are cross-checked with othe
  ] applies to pump_cmd;
**};

end PCA_Interlock_System.imp;
end PCA_Interlock_System;
```

What specific fault will result?

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

STPA in AADL

Where would the bad control action come from?



Integrated Hazard Analysis

Specification Step 1: Out Propagation

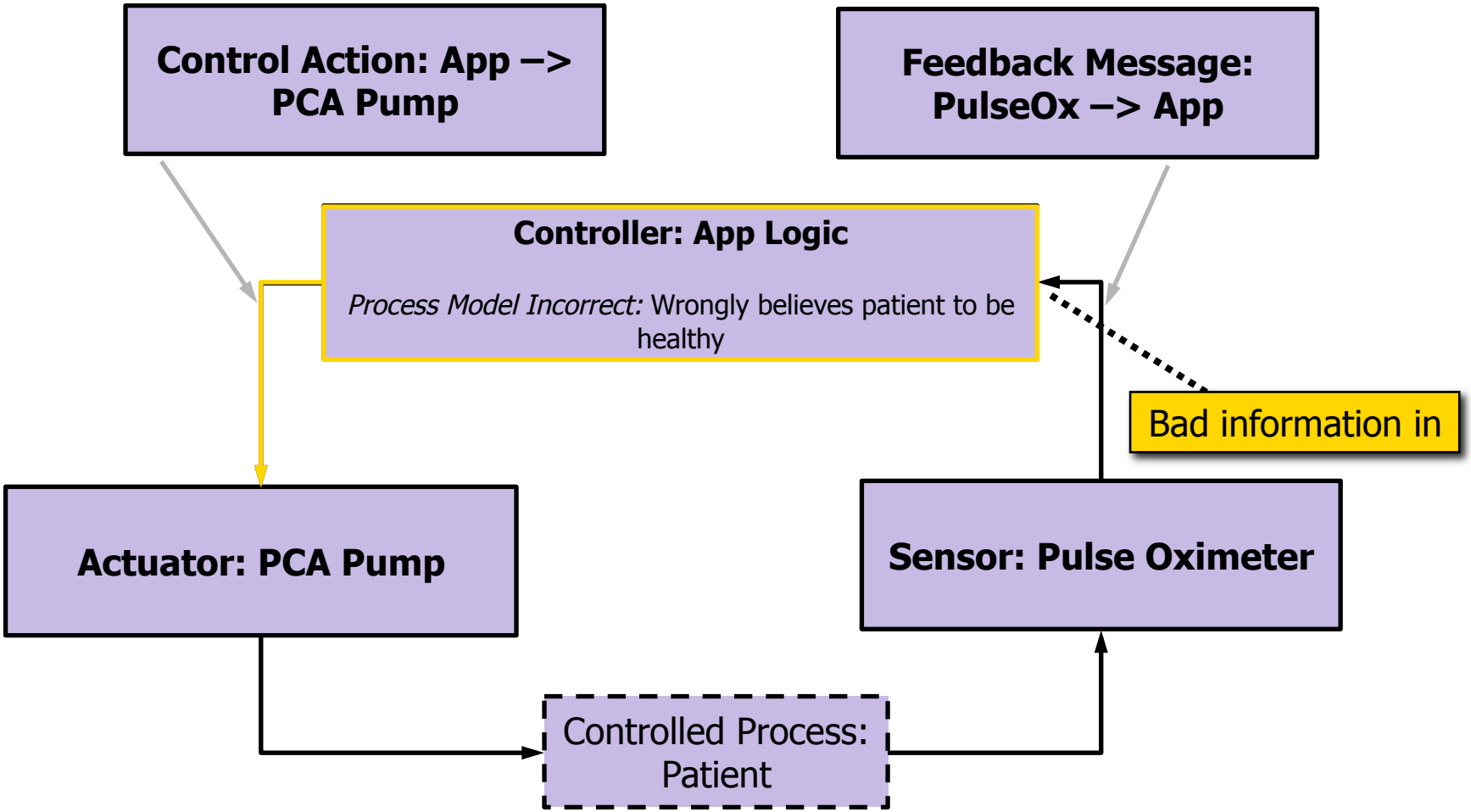

```
package PCA_Shutoff_Logic
public
with PCA_Shutoff_Types, PCA_Shutoff_Properties, MAP_Properties;

   process ICEpcaShutoffProcess
   features
      SpO2 : in event data port PCA_Shutoff_Types::SpO2;
      CommandPumpNormal : out event data port PCA_Shutoff_Types::PumpNormalCommand;
   properties
      MAP_Properties::Component_Type => logic;
   annex EMV2 {**
      use types PCA_Shutoff_Errors;
      error propagations
         SpO2 : in propagation {SpO2ValueHigh};
         CommandPumpNormal : out propagation {InadvertentPumpNormally};
         flows
            HighSpO2LeadsTOD : error path SpO2{SpO2ValueHigh} -> CommandPumpNormal{InadvertentPumpNormally};
      end propagations;
   **};
   end ICEpcaShutoffProcess;

   -- Process implementation redacted
end PCA_Shutoff_Logic;
```

Outgoing Port

Outgoing Fault

STPA in AADL

Where would the bad control action come from?



Integrated Hazard Analysis

Specification Step 2: In Propagation

Incoming Port

Incoming Fault

```
package PCA_Shutoff_
public
with PCA_Shutoff_Types, PCA_Shutoff_Properties, MAP_Properties;

   process ICEpcaShutoffProcess
   features
      SpO2 : in event data port PCA_Sh
      CommandPumpNormal : out event data port PCA_Shutoff_Types::PumpNormalCommand;
   properties
      MAP_Properties::Component_Type => logic;
   annex EMV2 {**
      use types PCA_Shutoff_Errors;
      error propagations
         SpO2 : in propagation {SpO2ValueHigh};
         CommandPumpNormal : out propagation {InadvertentPumpNormally};
         flows
            HighSpO2LeadsToOD : error path SpO2{SpO2ValueHigh} -> CommandPumpNormal{InadvertentPumpNormally};
      end propagations;
   **};
   end ICEpcaShutoffProcess;

   -- Process implementation redacted
end PCA_Shutoff_Logic;
```

Integrated Hazard Analysis

Specification Step 3: Relation between incoming and outgoing


```
package PCA_Shutoff_Logic
public
with PCA_Shutoff_Types, PCA_Shutoff_Properties, MAP_Properties;

   process ICEpcaShutoffProcess
   features
      SpO2 : in event data port PCA_Shutoff_Types::SpO2;
      CommandPumpNormal : out event data port PCA_Shutoff_Types::PumpNormal
   properties
      MAP_Properties::Component_Type =>
   annex EMV2 {**
      use types PCA_Shutoff_Errors;
      error propagations
         SpO2 : in propagation {SpO2ValueHigh};
         CommandPumpNormal : out propagation {InadvertentPumpNormally};
         flows
            HighSpO2LeadsToOD : error path SpO2{SpO2ValueHigh} -> CommandPumpNormal{InadvertentPumpNormally};
      end propagations;
   **};
   end ICEpcaShutoffProcess;

   -- Process implementation redacted
end PCA_Shutoff_Logic;
```

Name of flow

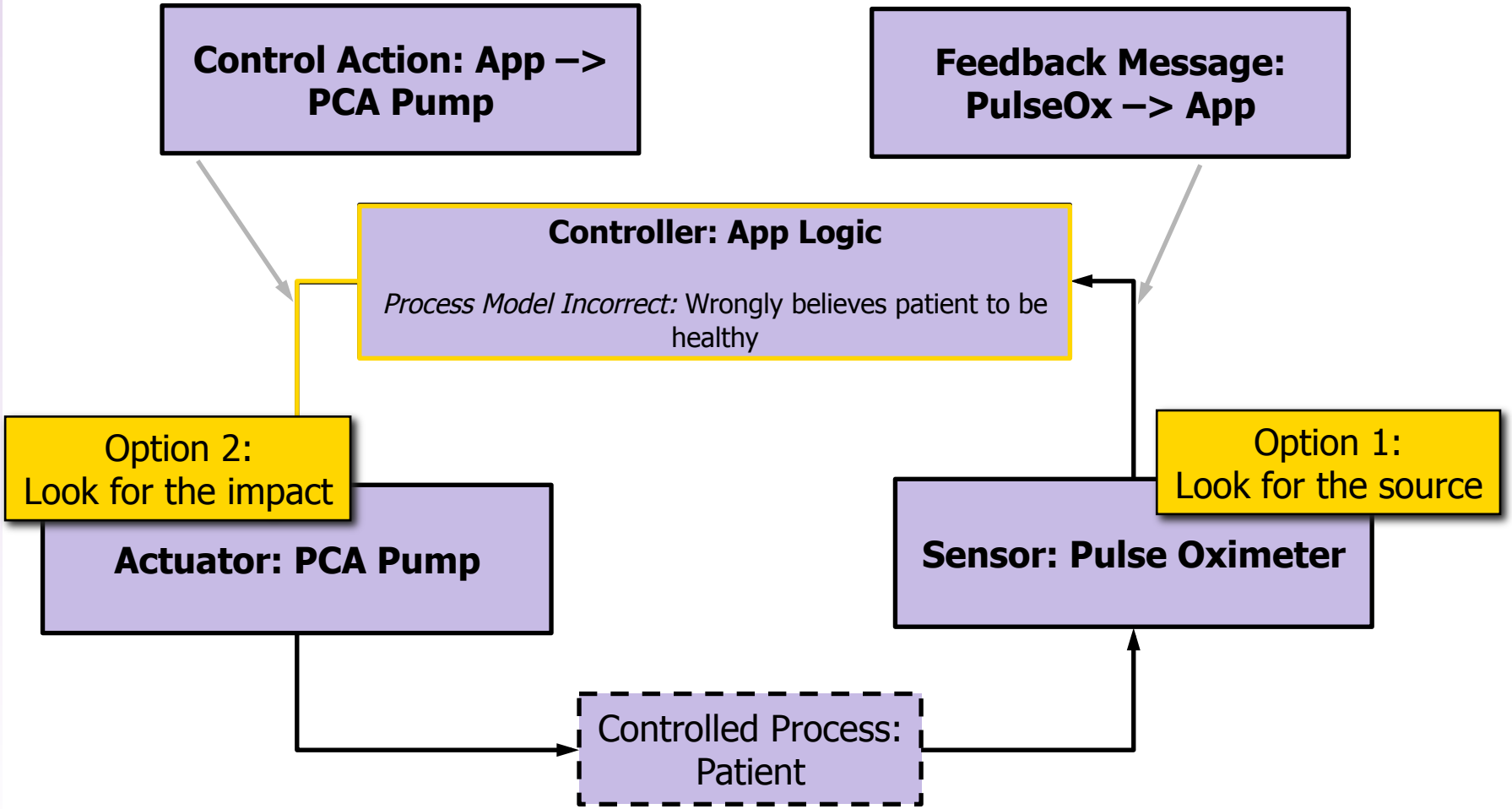
Type of flow

Specific faults

Specific Ports

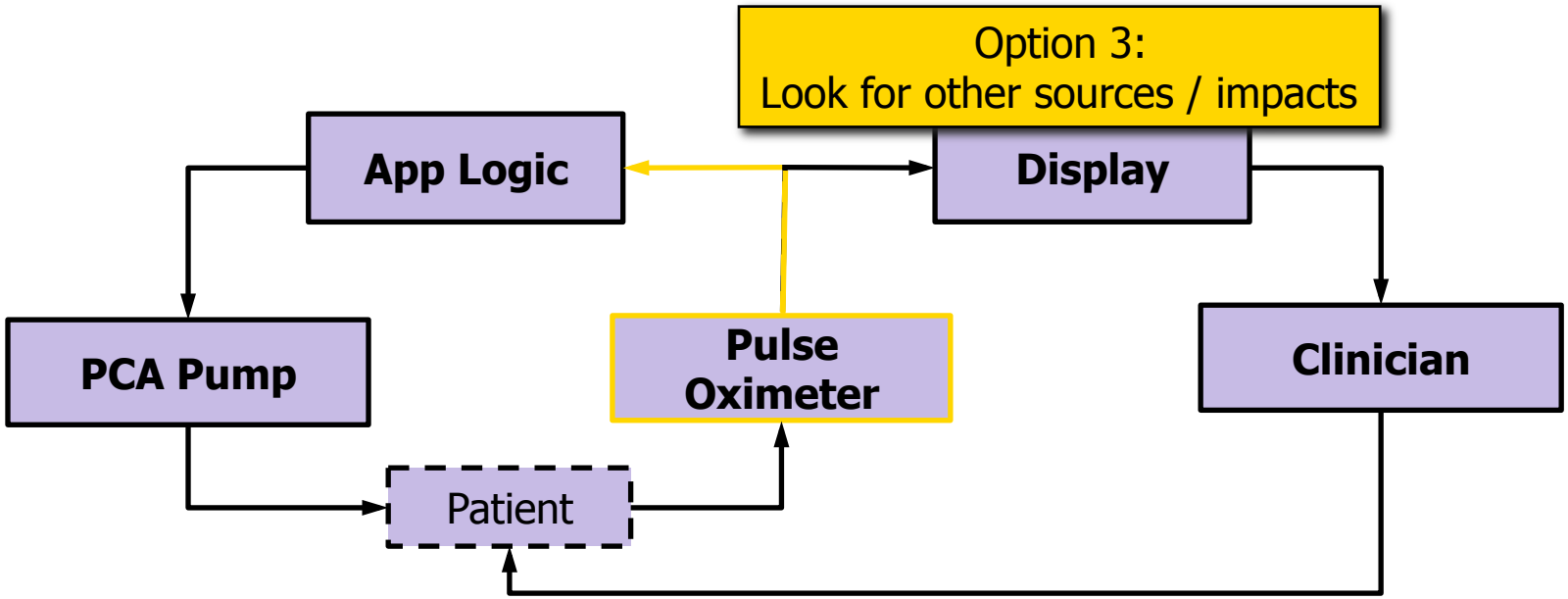
STPA in AADL

Where should we go now?



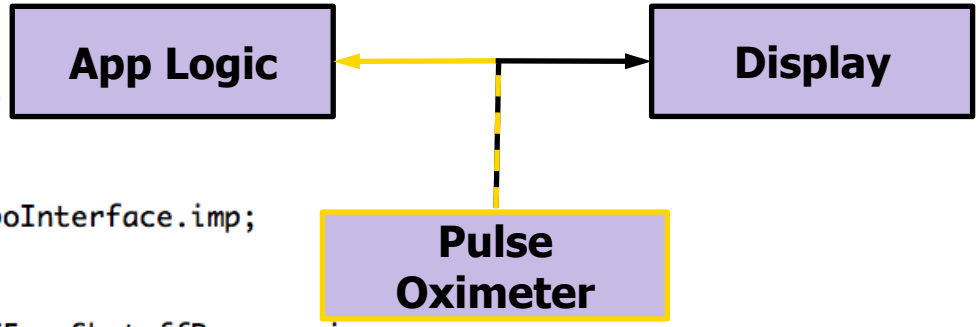
STPA in AADL

Where should we go now?



Integrated 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;
annex
```

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

```
    -- 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";
```

Tool Supported Process

Interaction between Report and Model

Cause –> Effect

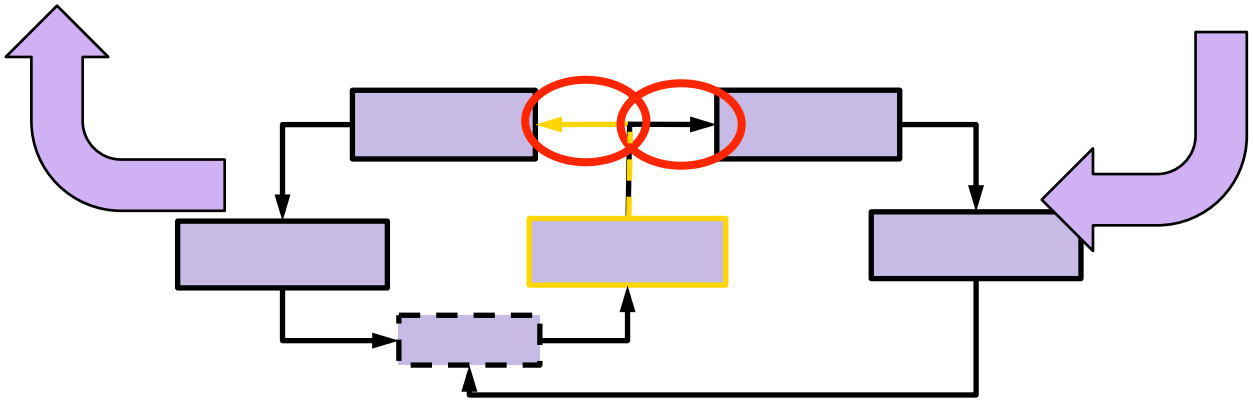
CONTROL ACTION	PROVIDING	NOT PROVIDING	APPLIED TOO LONG	STOPPED TOO SOON	EARLY	LATE
spo2_disp	H2 (Wrong Values (Undetected))					
pulseox_fail_disp						
etco2_logic						
pumpcommand_disp						
respiratoryrate_logic	H1 (Wrong values (Detected)), H1 (Wrong values (Detection Dropped))		H1 (Network Drop)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

1. Here's an empty cell (STPA Keyword + Control Action)… could anything go wrong?

2. Create occurrence and supporting EM annotations

3. Where else could this fault go?

4. What else could cause this error?



Effect –> Cause

Further Reading

- Source available online at https://github.com/santoslab/aadl-translator
- Installable into OSATE2 via update site: http://santoslab.org/pub/mdcf-architect/updatesite
- Full documentation online at http://santoslab.org/pub/mdcf-architect
- Publications online at http://people.cis.ksu.edu/~samprocter

Using STPA to Support Risk Management for Interoperable Medical Systems

STAMP Workshop 2015, MIT

Sam Procter, John Hatcliff
SAnToS Lab
Kansas State University

Anura Fernando
Underwriters Laboratories

Sandy Weininger
US Food and Drug Administration

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.

Referee Comments

- Doesn’t use of AADL imply a fully specified architecture?
  - No. Though some architectural constraints are implied by the domain (eg, component-based architecture, use of underlying middleware for communication, etc.), architectures in AADL can be rapidly modified. Constructing (or modeling) an architecture in AADL is very much a “design phase” task.

Referee Comments

- How can apps be certified independently of their environment?
  - Much the same way that medical devices are currently certified under some set of assumptions (collectively referred to as *intended use*), we imagine that MAP apps will have (contra)indications for use
  - There are requirements engineering issues to be addressed, this is a key part of the UL 2800 standardization effort

Referee Comments

- What about interactions between devices / apps that are not over input or output ports?
  - We rely heavily on a notion of platform to isolate components from one another. This platform technology, developed by our King et al at UPenn, aims to provide complete separation between components (similar to separation kernels / partitioning middleware used in avionics)
  - AADL can also model unintended / indirect interactions, like heat