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

- Information Systems
- Sensors
- Actuators
- Sensor Data Displays
- Clinical Protocols
- Clinicians
- Information Systems
- Patient
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

Devices

- PCA Pump
  - Enable Pump for safe time window
- Capnograph
  - Monitoring Data + Alarm Information
- Pulse Oximeter
  - Monitoring Data + Alarm Information

Device Task controller
- Enable bolus dose only when ticket present
- PCA Bolus "Enable" Ticket
- Aggregated Monitoring Status

Combined PCA Vitals Monitoring

Clinician / Monitoring
- Status Display for PCA Monitoring Application
Device1

Output rate: 1 sec .. 5 sec

Device2

AADL System

AADL Process: Logic

Thread1

Thread2

Thread3

AADL Process: Display

Thread1

Thread2

Channel Delay: 50ms

Period: 50ms

WCET: 5ms
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."
];
```
STPA

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

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Providing</th>
<th>Not Providing</th>
<th>Applied too Long</th>
<th>Stopped too Soon</th>
<th>Early</th>
<th>Late</th>
</tr>
</thead>
<tbody>
<tr>
<td>App -&gt; Pump: Pump Normally</td>
<td>PH</td>
<td>Not Hazardous</td>
<td>PH</td>
<td>Not Hazardous</td>
<td>PH</td>
<td>Not Hazardous</td>
</tr>
<tr>
<td>App -&gt; Disp: Patient Ok</td>
<td>BID</td>
<td>BID</td>
<td>BID</td>
<td>BID</td>
<td>BID</td>
<td>BID</td>
</tr>
<tr>
<td>PulseOx-&gt;App: Provide SpO₂</td>
<td>Not Hazardous</td>
<td>PH, BID</td>
<td>Not Hazardous</td>
<td>PH, BID</td>
<td>Not Hazardous</td>
<td>PH, BID</td>
</tr>
<tr>
<td>PulseOx-&gt;App: Provide Pulse Rate</td>
<td>Not Hazardous</td>
<td>PH, BID</td>
<td>Not Hazardous</td>
<td>PH, BID</td>
<td>Not Hazardous</td>
<td>PH, BID</td>
</tr>
</tbody>
</table>
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
Feedback or control action is provided in an unsafe way

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?
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: ICEpointInterface;
  applogic : process PCA_Shutoff_Logics: 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 port empty";
      Compensation => "Physiological readings have a maximum valid time, it is no longer 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

Annotating our Architectural Model

Inside the AADL System Component

```plaintext
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_Lo
```
Hazard Analysis

Annotating the Architectural Model

The fault is traced to its source component / port
Hazard Analysis

Specification Step 1: Propagation

```latex
package PulseOx_Interface
public
with PCA_Shutoff_Types, PCA_Shutoff_Errors, EMV2, MAP, Error_Properties, PCA_Shutoff;

device ICEpoInterface
features
  Sp02 : out event data port PCA_Shutoff_Types::Sp02;
annex EMV2 {**
  use types PCA_Shutoff_Errors;
  error propagations
    Sp02 : out propagation {Sp02ValueHigh};
  flows
    Sp02UndetectableHighValueFlowSource : error source Sp02 {Sp02ValueHigh};
  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

Anything missing?

There are two missed error propagations!
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;

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

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.