STPA Analysis of Intravenous Patient-Controlled Analgesia

John Thomas

Ying Hann Ang

Kristie Chung

Olivia Qing Gao

Intravenous Patient-Controlled Analgesia (IV-PCA)

- Allows self-administration of analgesia for pain management
- Safety features
  - Pump delay
  - Lock out interval
  - Limit on total dosage per time interval
- 56,000 FDA reports of problems caused by drug infusion pumps
- 500 deaths from 2005-2009**
  - About 67% due to equipment failure
  - About 33% are operator-related
- 60% of pump infusions contained some type of error*
  - 65% of errors due to violations of labeling and tubing change policies
- 87 PCA infusion pumps recalled due to safety issues

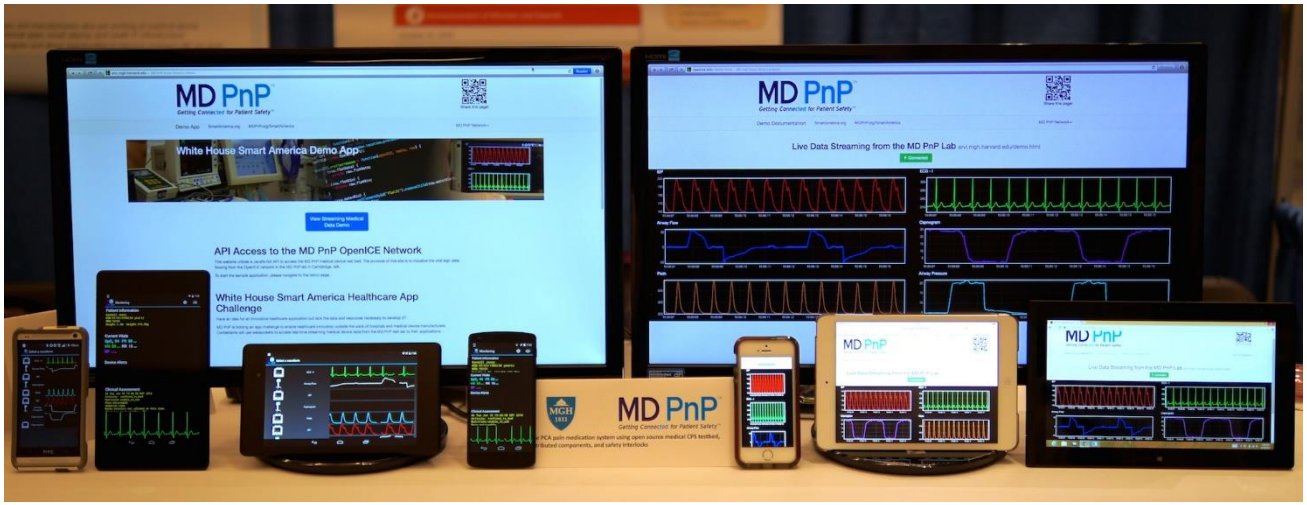
*Schnock, et al. 2016
**Hankin, et al. 2007

Historical problems with PCA pumps

- Equipment failures
  - E.g. failures in tube, clamp, syringe barrel, seal, etc.
- Misprogramming
  - 19-year old woman underwent a C-section, killed by overdose due to a programming error that mismatched drug dosage with drug cassette type (Vicente et al.)
- Over-sedation
  - Extreme over-sedation, respiratory depression, respiratory arrest
  - By proxy: US Pharmacopeia reported at least 15 cases
- Drug mix-up
  - E.g. hydromorphone mixed up with morphine

Integrated Clinical Environment (ICE)

- Allows interoperability between medical devices
- Allows new safety interlock systems
  - Read from pulse oximeter
  - Read from capnogram
  - Detect patient condition
  - Alert clinicians about a patient in distress
  - Stop drug administration



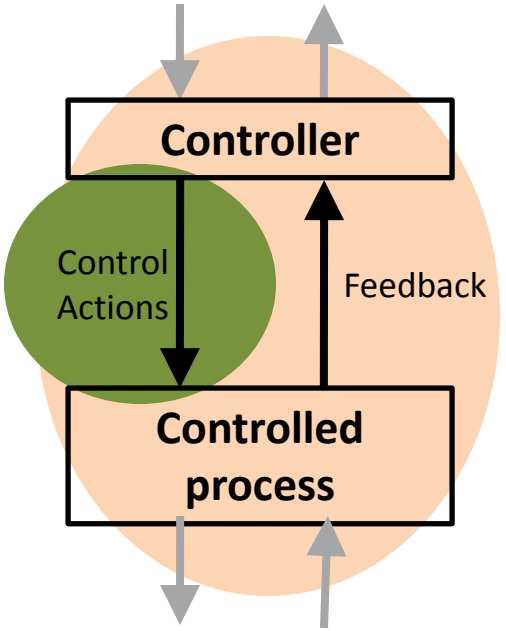
MD PnP OpenICE

STPA
(System-Theoretic Process Analysis)

STPA Hazard Analysis

STAMP Model

- Identify accidents and hazards
- Draw the control structure
- Step 1: Identify unsafe control actions
- Step 2: Identify causal scenarios



Can capture software problems, human errors, design flaws, etc.

(Leveson, 2012)

STPA: Identify accidents and hazards

Accidents (Losses)

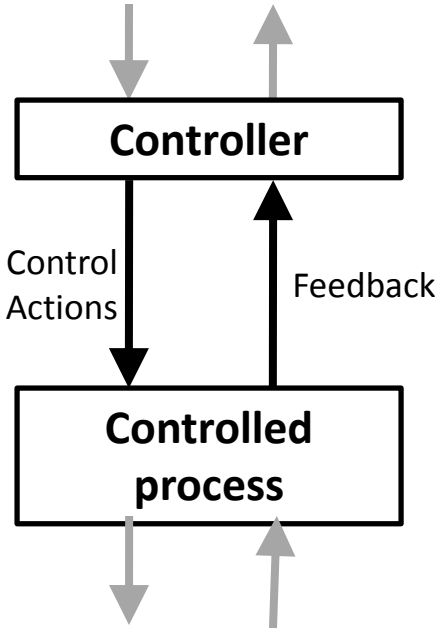
- **A1:** Loss of life or serious injury to patient
- **A2:** Patient's pain is not relieved
- **A3:** Loss of protected patient or proprietary hospital information
- **A4:** Financial loss or loss of hospital reputation

System Hazards

- **H1**: Opioid overdose **[A1, A4]**
- **H2:** Opioid under-dose **[A2]**
- **H3:** Unauthorized access to hospital/patient information **[A3, A4]**

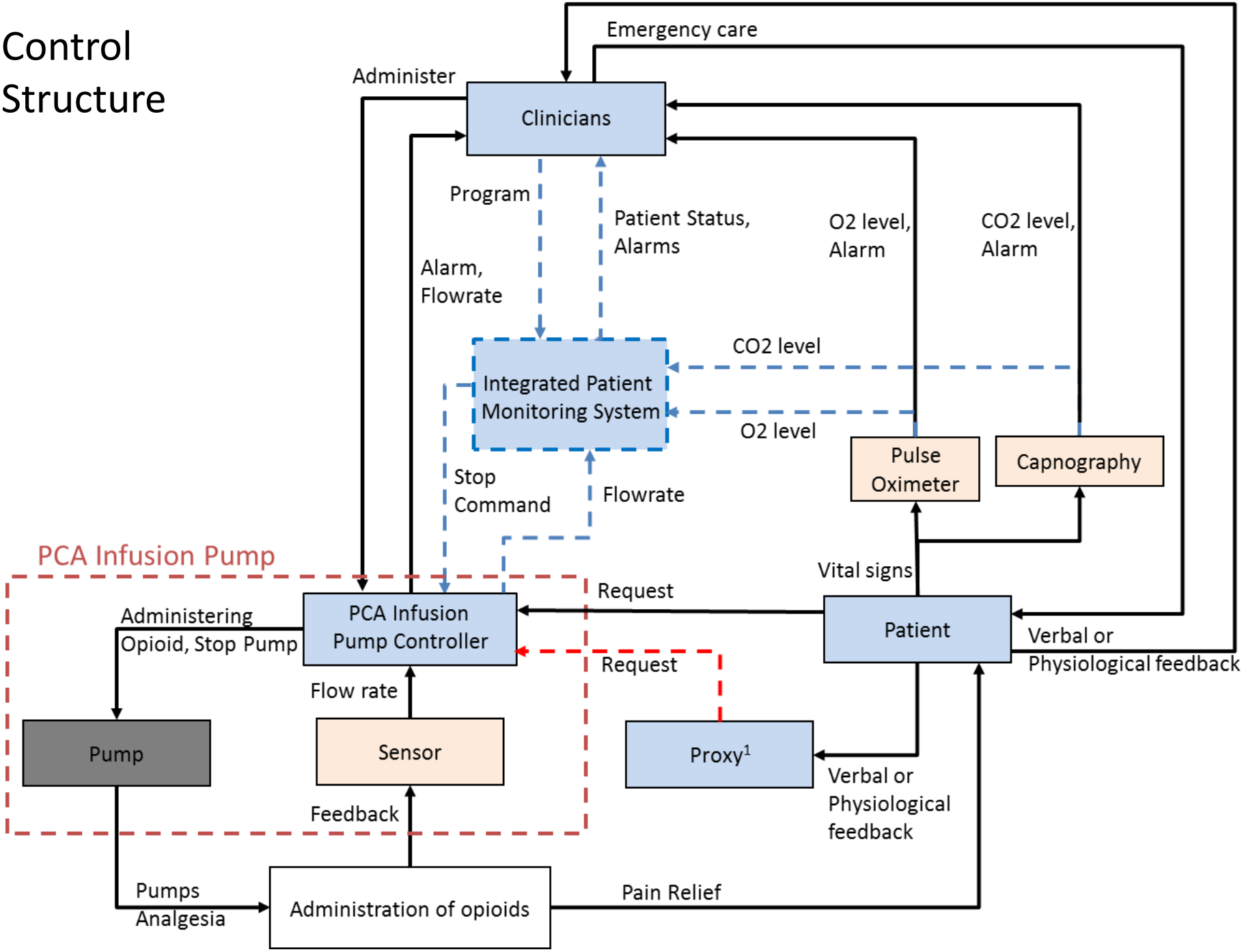
STPA
(System-Theoretic Process Analysis)

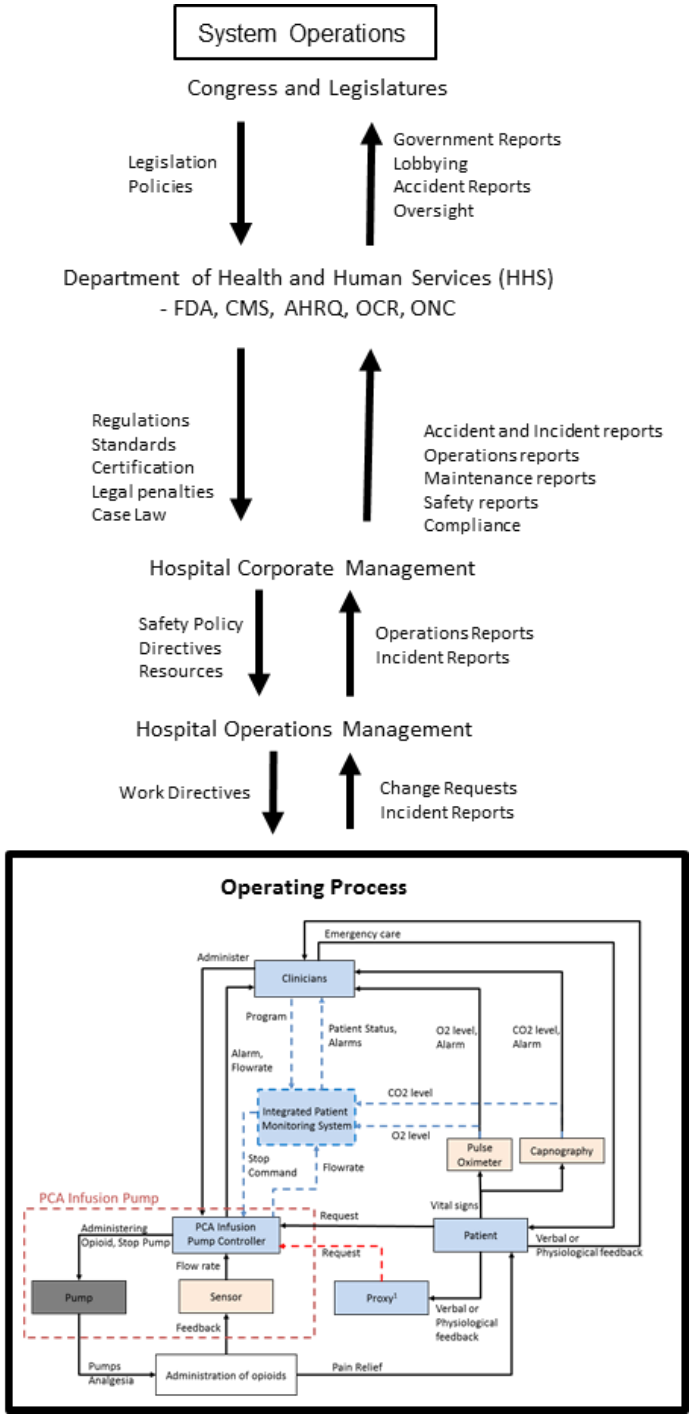
- Identify accidents and system hazards
- Draw the control structure
- Step 1: Identify unsafe control actions
- Step 2: Identify causal scenarios



(Leveson, 2012)

Control Structure

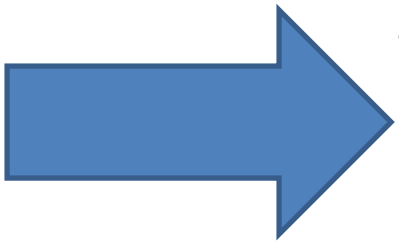



System Operations
Congress and Legislatures
Legislation
Policies
Government Reports
Lobbying
Accident Reports
Oversight
Department of Health and Human Services (HHS)
- FDA, CMS, AHRQ, OCR, ONC
Regulations
Standards
Certification
Legal penalties
Case Law
Accident and Incident reports
Operations reports
Maintenance reports
Safety reports
Compliance
Hospital Corporate Management
Safety Policy
Directives
Resources
Operations Reports
Incident Reports
Hospital Operations Management
Work Directives
Change Requests
Incident Reports
Operating Process
Emergency care
Administer
Clinicians
Program
Alarm, Flowrate
Patient Status, Alarms
O2 level, Alarm
CO2 level, Alarm
CO2 level
Integrated Patient Monitoring System
O2 level
Pulse Oximeter
Capnography
Stop Command
Flowrate
PCA Infusion Pump
Vital signs
Request
Administering Opioid, Stop Pump
PCA Infusion Pump Controller
Patient
Verbal or Physiological feedback
Flow rate
Request
Pump
Sensor
Proxy[1]
Verbal or Physiological feedback
Feedback
Pumps Analgesia
Administration of opioids
Pain Relief

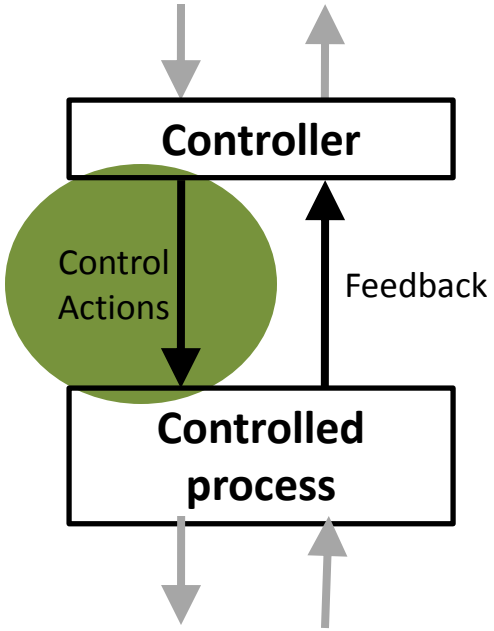
STPA
(System-Theoretic Process Analysis)

- Identify accidents and hazards

- Draw the control structure

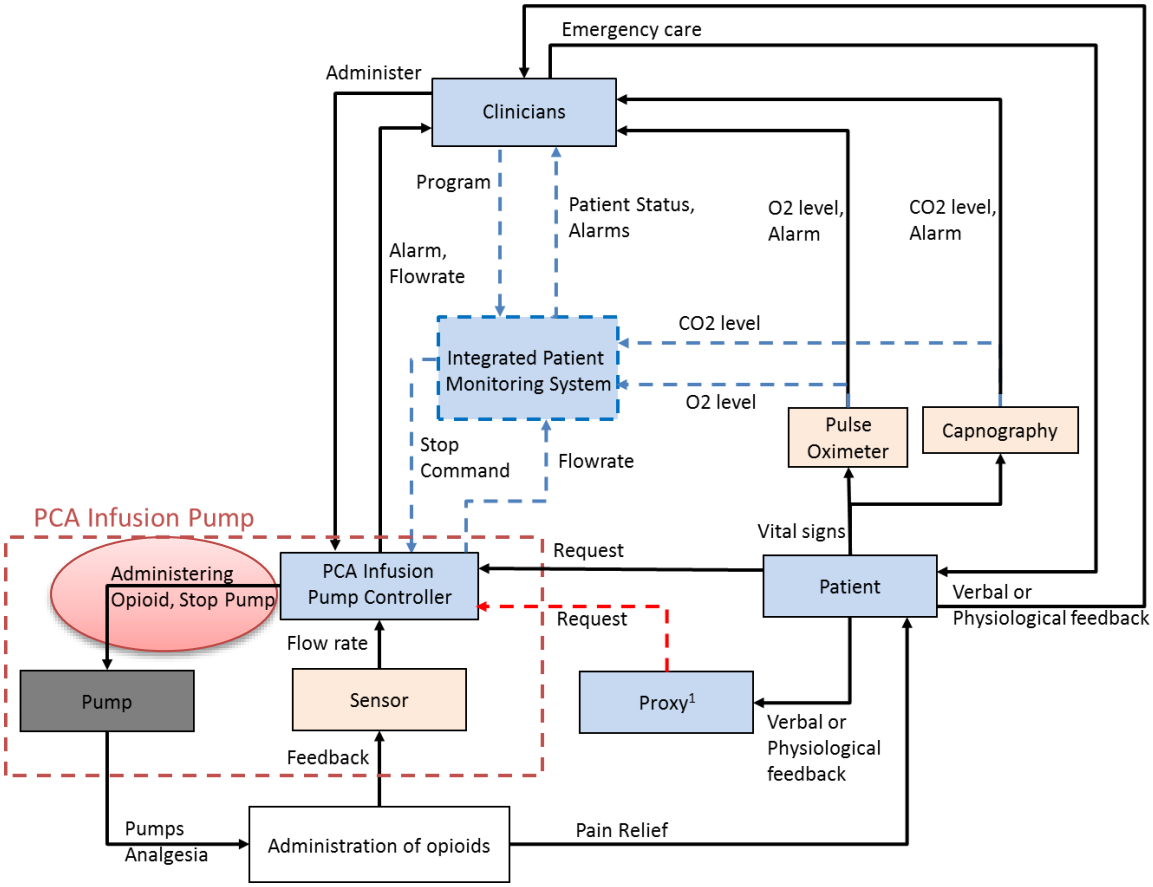


- **Step 1: Identify unsafe control actions**
- Step 2: Identify causal factors and create scenarios



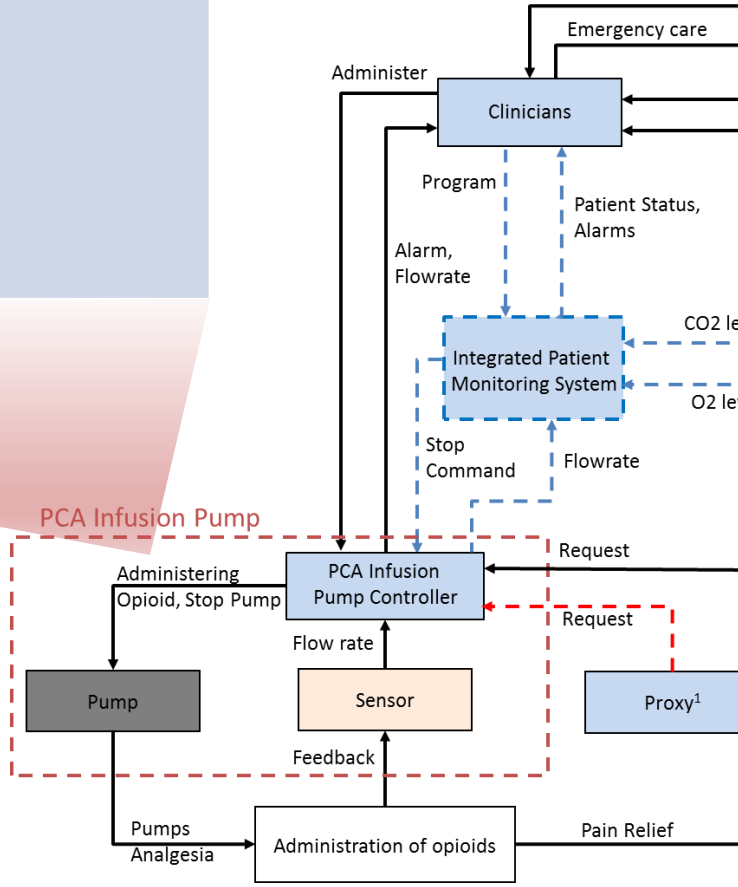
(Leveson, 2012)

Unsafe Control Actions



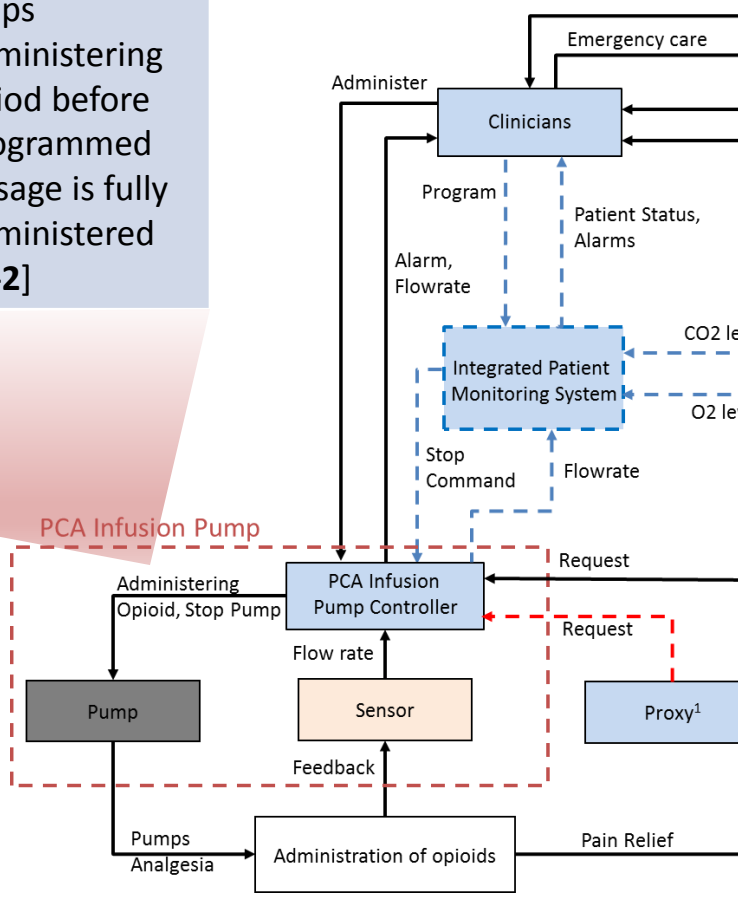
Unsafe Control Actions

Control Action	Not providing causes hazard	Providing causes hazard	Wrong timing/Order causes hazard	Stopped too Soon/Applied Too Long
Administer Opiod				



Unsafe Control Actions

Control Action	Not providing causes hazard	Providing causes hazard	Wrong timing/Order causes hazard	Stopped too Soon/Applied Too Long
Administer Opiod	UCA-PCA-1: PCA controller does not administer opiod when patient is in pain **[H-2]**	UCA-PCA-2: PCA controller administers opiod when patient has already been given too much opiod **[H-1]**	UCA-PCA-3: Delay in PCA controller administering opioid beyond TBD seconds/minutes **[H-2]**	UCA-PCA-4: PCA controller stops administering opiod before programmed dosage is fully administered [**H-2**]

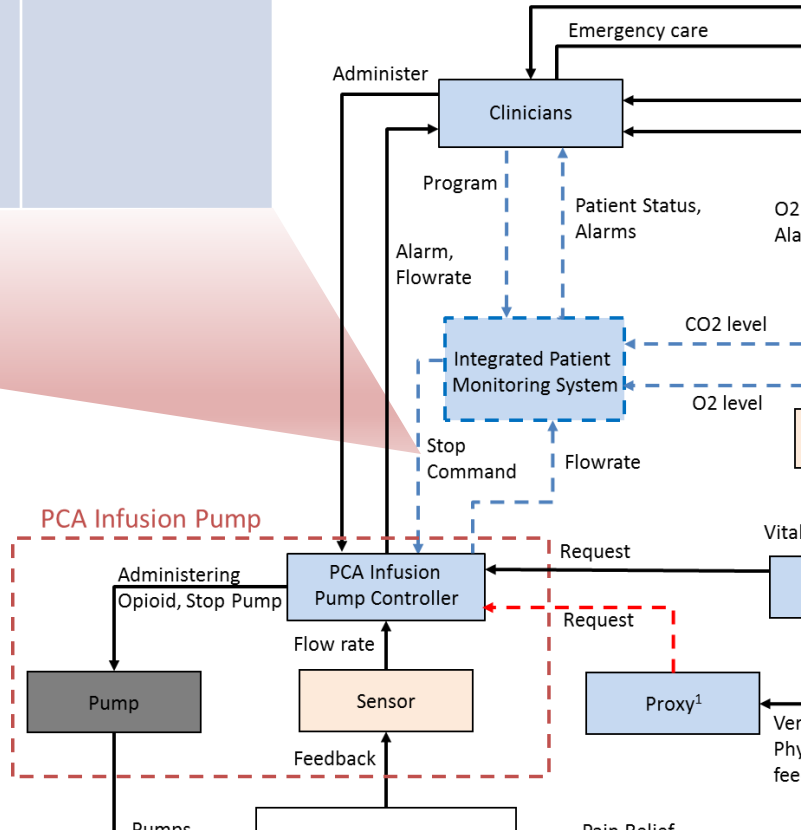


Hazards

- **H1**: Opioid overdose **[A1, A4]**
- **H2:** Opioid under-dose **[A2]**
- **H3:** Unauthorized access to hospital/patient information **[A3, A4]**

Unsafe Control Actions

Control Action	Not providing causes hazard	Providing causes hazard	Wrong timing/Order causes hazard	Stopped too Soon/Applied Too Long
Stop Command	UCA-IPMS-01. IPMS does not send a STOP command to PCA when patient has been given too much opioid **[H-1]**	UCA-IPMS-02. IPMS sends a STOP command to PCA when there is no emergency. **[H-2]**	UCA-IPMS-03. IPMS takes too long to send a STOP command in an emergency. **[H-1]**	N/A

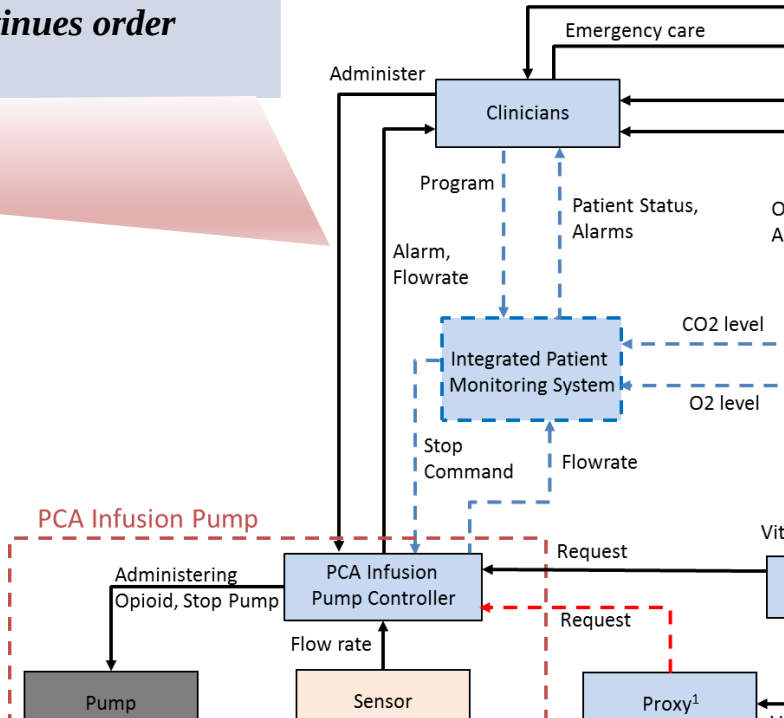


Hazards

- **H1**: Opioid overdose **[A1, A4]**
- **H2:** Opioid under-dose **[A2]**
- **H3:** Unauthorized access to hospital/patient information **[A3, A4]**

Unsafe Control Actions

Control Action	Not providing causes hazard	Providing causes hazard	Wrong timing/Order causes hazard	Stopped too Soon/Applied Too Long
Administer PCA pump	UCA-CLN-01. Clinician does not administer PCA pump per prescription/instruction **[H-1, H-2]**	UCA-CLN-02. Clinician administers PCA pump when patient is not in need **[H-1]**	UCA-CLN-03. Clinician administers PCA pump too late **[H-2]**	UCA-CLN-04. Clinician administration of PCA is interrupted (stopped too soon) **[H-2]** ***UCA-CLN-05. Clinician administers PCA too long after physician discontinues order [H-1]***

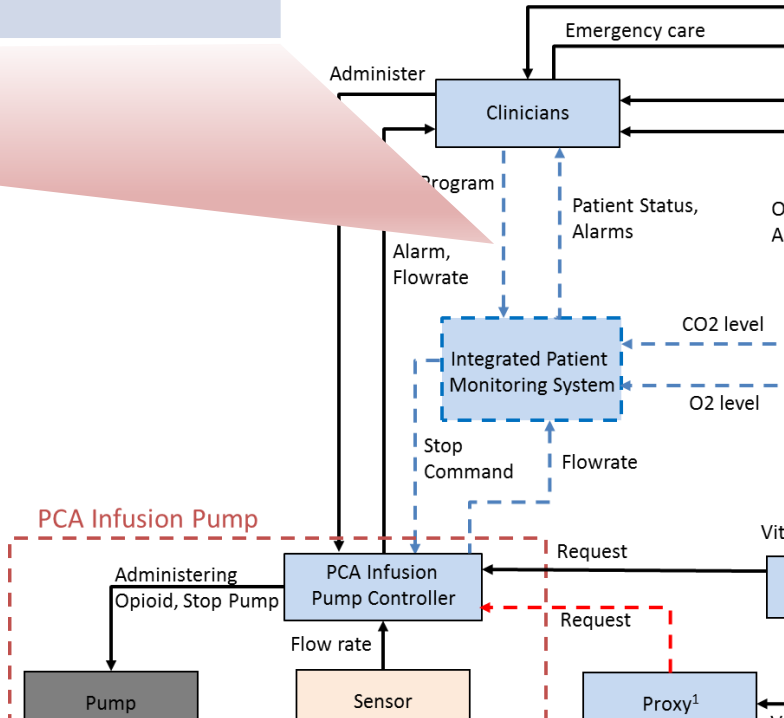


Hazards

- **H1**: Opioid overdose **[A1, A4]**
- **H2:** Opioid under-dose **[A2]**
- **H3:** Unauthorized access to hospital/patient information **[A3, A4]**

Unsafe Control Actions

Control Action	Not providing causes hazard	Providing causes hazard	Wrong timing/Order causes hazard	Stopped too Soon/Applied Too Long
Program IPMS	UCA-CLN-06. Clinician does not program IPMS prior to use **[H-1, H-2]**	UCA-CLN-07. Clinician programs IPMS with incorrect value **[H-1, H-2]**	UCA-CLN-06. Clinician programs IPMS too late while it is already in use **[H-1, H-2]**	N/A



Hazards

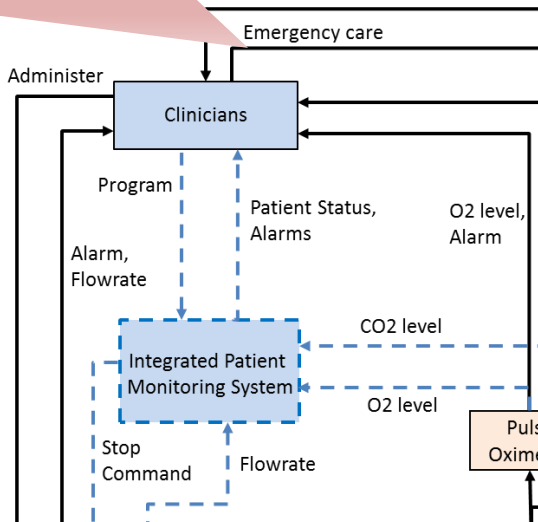
- **H1**: Opioid overdose **[A1, A4]**
- **H2:** Opioid under-dose **[A2]**
- **H3:** Unauthorized access to hospital/patient information **[A3, A4]**

Unsafe Control Actions

Control Action	Not providing causes hazard	Providing causes hazard	Wrong timing/Order causes hazard	Stopped too Soon/Applied Too Long
Provide PCA handling instructions	UCA-CLN-08. Clinician does not provide instructions to patient or provided in a way that patient does not understand (e.g., language or culture issues) **[H-1, H-2]**	UCA-CLN-09. Clinician provides incorrect instructions to patient **[H-1, H-2]**	UCA-CLN-10. Clinician provides instructions too soon or too late **[H-1, H-2]**	

Hazards

- **H1**: Opioid overdose **[A1, A4]**
- **H2:** Opioid under-dose **[A2]**
- **H3:** Unauthorized access to hospital/patient information **[A3, A4]**

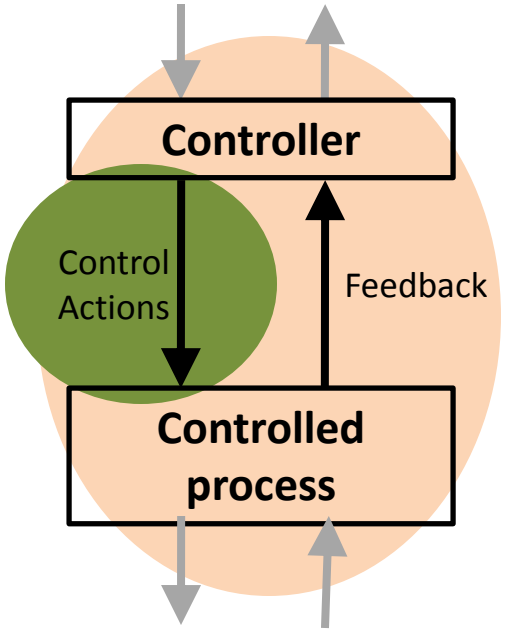


Creating safety constraints

Unsafe Control Actions	Safety Constraints
UCA-IPMS-01 IPMS does not send a STOP command to PCA when patient has been given too much opioid **[H-1]**	SC-1 IPMS must send STOP command to PCA when patient has been given too much opiod
UCA-IPMS-02 IPMS sends a STOP command to PCA when there is no emergency **[H-2]**	IPMS must not sent STOP command unless an emergency condition is indicated
UCA-IPMS-03 IPMS takes too long to send a STOP command in an emergency **[H-1]**	IPMS must sent STOP command within TBD seconds of emergency
Etc.	Etc.

STPA
(System-Theoretic Process Analysis)

- Identify accidents and hazards
- Draw the control structure
- Step 1: Identify unsafe control actions
- Step 2: Identify causal scenarios



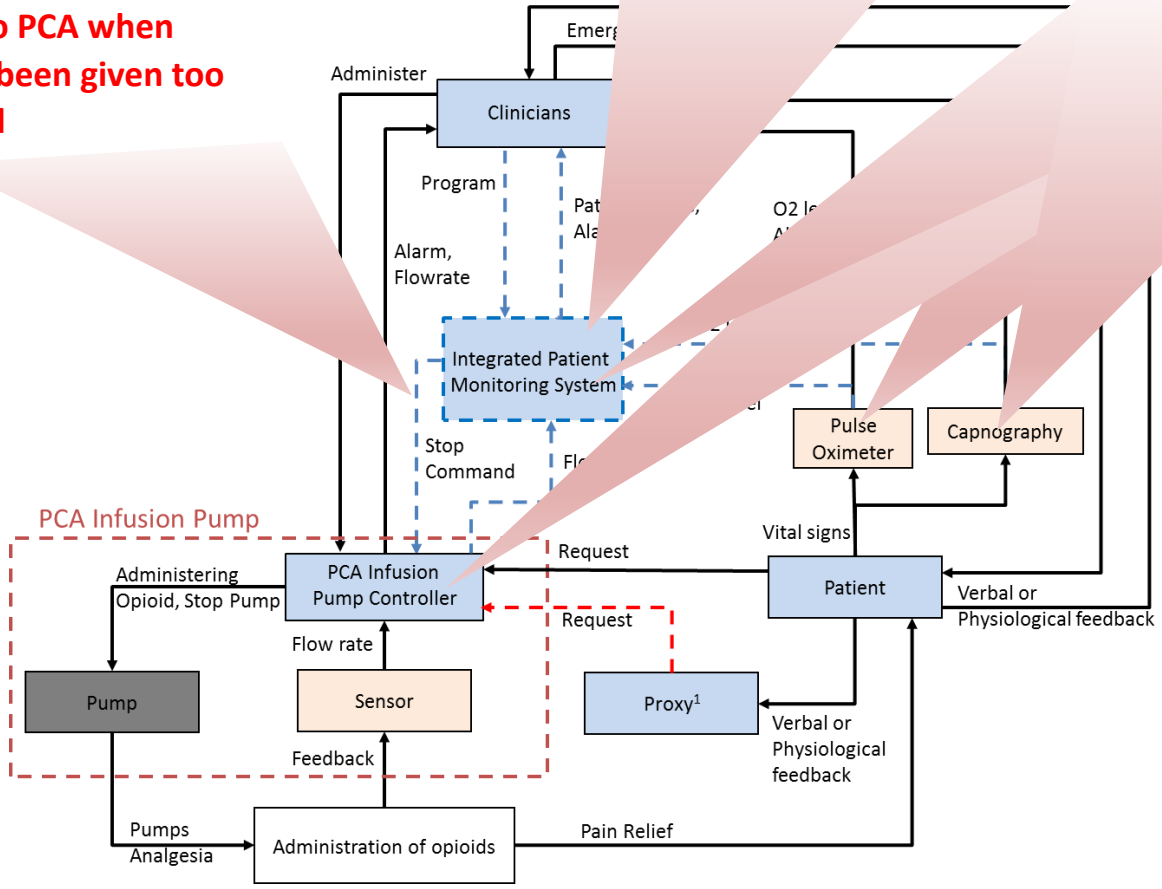
(Leveson, 2012)

Identify causal scenarios

UCA-IPMS-1:
IPMS does not send a STOP command to PCA when patient has been given too much opioid

IPMS does not believe patient is in overdose condition

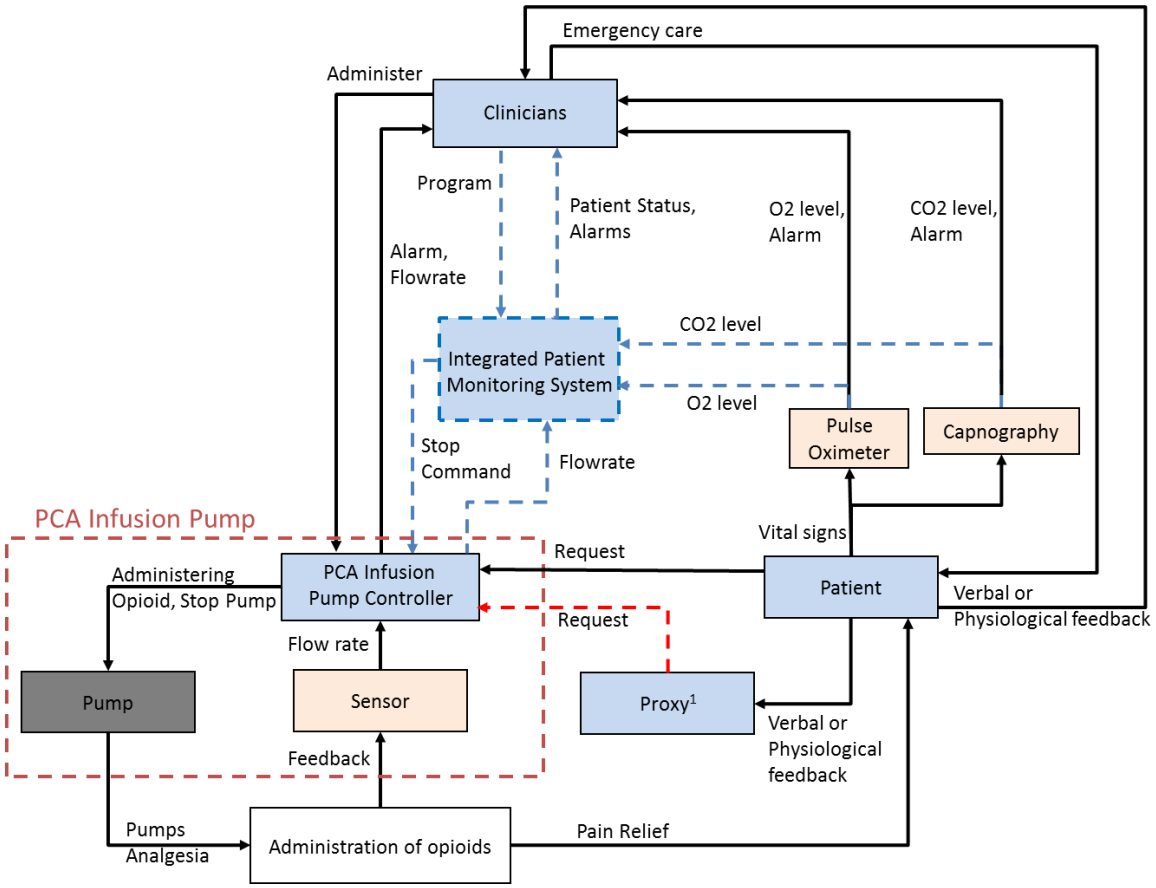
Messages discarded due to clock drift / incorrect time synchronization



Identify causal scenarios

Potential solutions:

- Implement time synchronization function in IPMS (e.g. Berkeley algorithm, NTP, etc.)
- Implement periodic synchronization checks

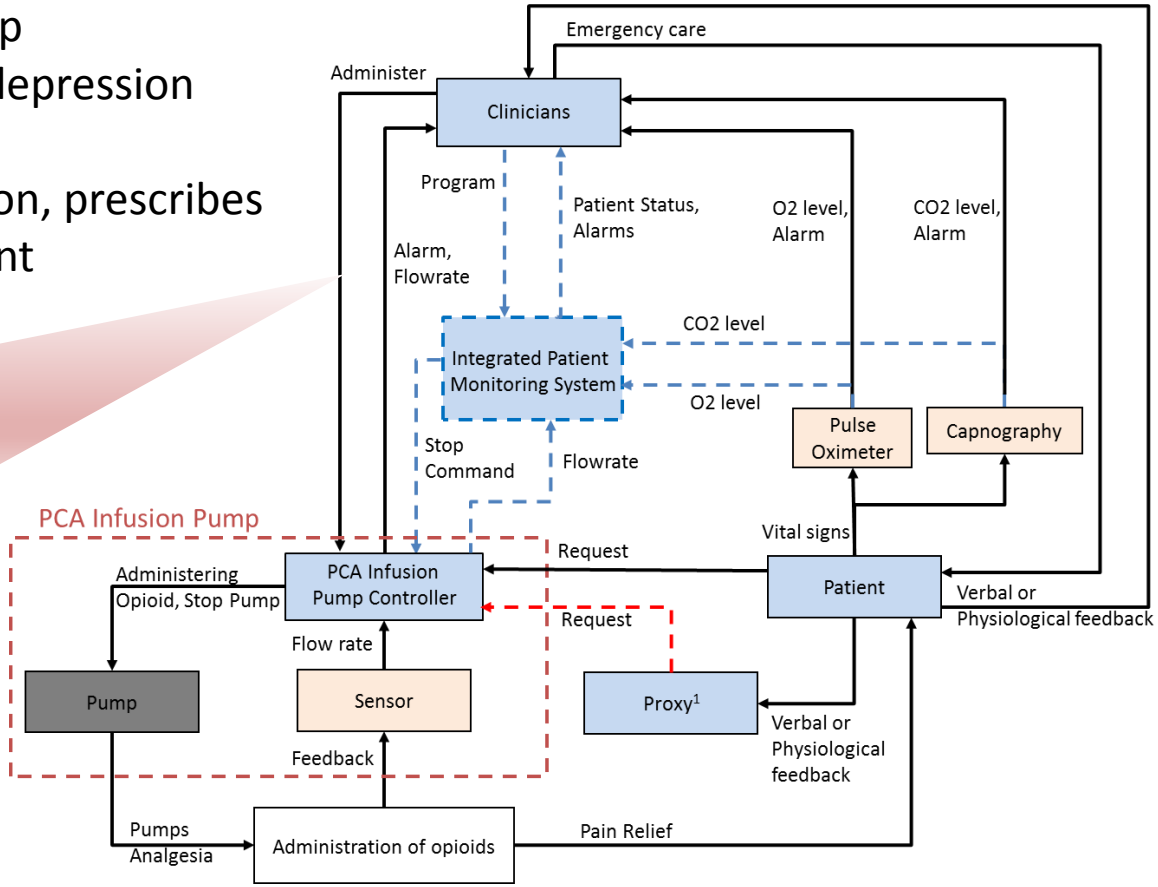


Identify causal scenarios

Clinician Scenario #9:

- PCA pump properly programmed
- Patient goes into respiratory depression.
- Clinician on day shift responds, removes PCA pump, overrides the alarm.
- Clinician shift change
- Patient complains about pain
- Clinician reconnects PCA pump
- Patient goes into respiratory depression for second time
- Physician not aware of situation, prescribes stronger oral pain management

UCA-CLN-02. Clinician administers PCA pump when it is not appropriate



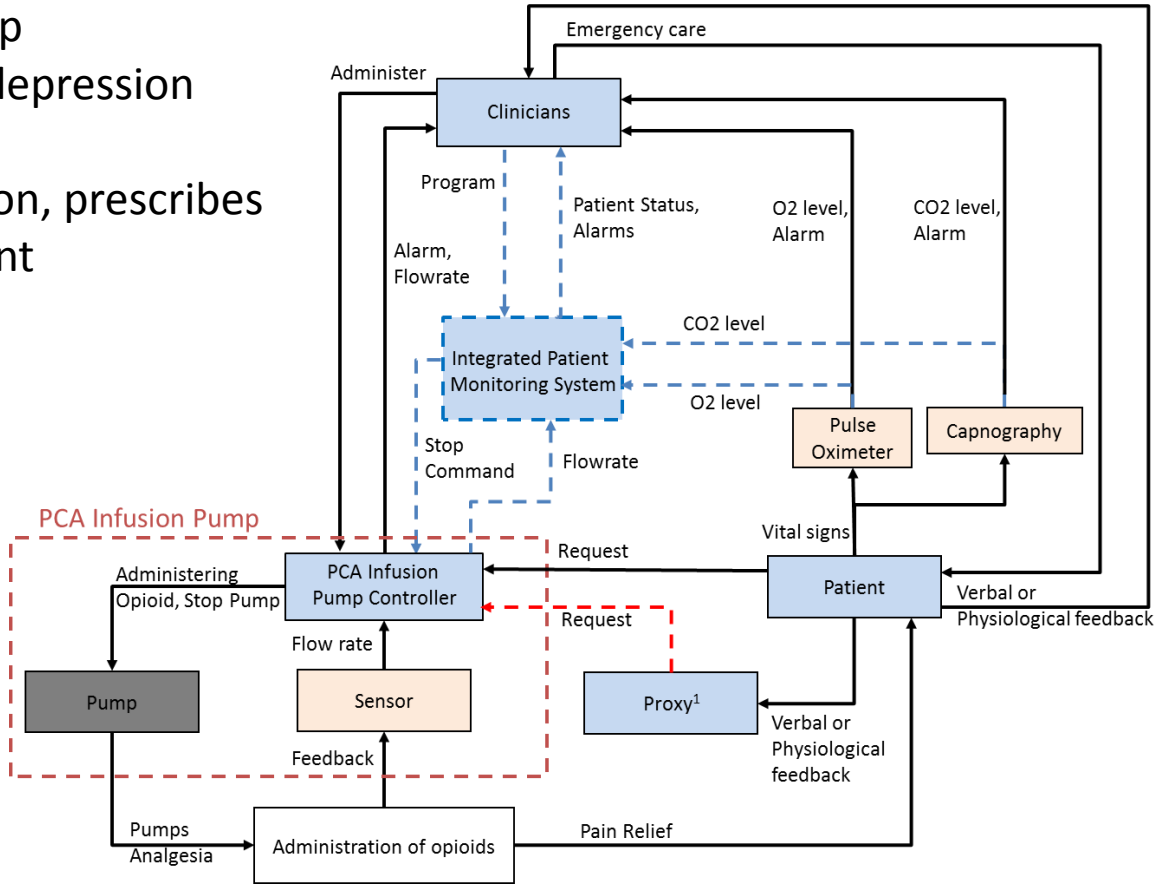
Identify causal scenarios

Clinician Scenario #9:

- PCA pump properly programmed
- Patient goes into respiratory depression.
- Clinician on day shift responds, removes PCA pump, overrides the alarm.
- Clinician shift change
- Patient complains about pain
- Clinician reconnects PCA pump
- Patient goes into respiratory depression for second time
- Physician not aware of situation, prescribes stronger oral pain management

Potential solutions:

- Procedures regarding PCA event communication, documentation
- IPMS could automatically record emergency incidents, further updated in the EHR system



Conclusions

- Captured potential human errors, software problems, interoperability issues
- Captures system-level behavior, context
- Captures interaction accidents without failures
  - Even when system operates *exactly as designed*
- Can be used to help develop solutions
  - Prevent unsafe scenarios
  - Ensure unsafe scenarios are observable and reversible
  - Detect missing feedback loops, e.g. operators and product designers
- Can be applied to organizational and social aspects