Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Assuring the Safety, Security, and Reliability of Medical-Device Cyber-Physical Systems (MDCPS)

Insup Lee
PRECISE Center
Department of Computer and Information Science
University of Pennsylvania

November 16, 2012

MD PnP™
Getting connected for patient safety™

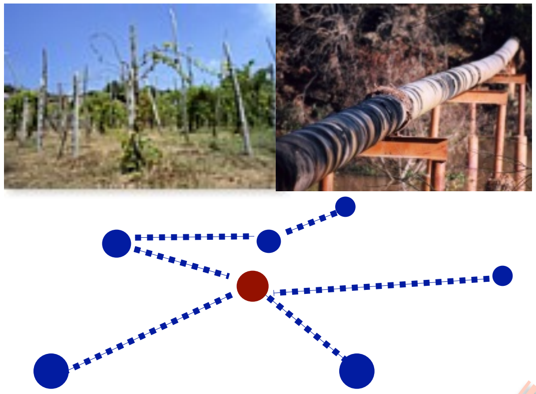
Penn Medicine

Trend 1: Device Proliferation

Embedded Everywhere!

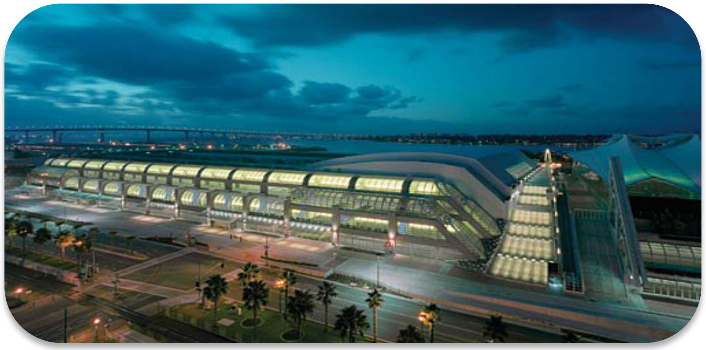
Smart Space

Trend 2: Integration at Scale



World Wide Sensor Web

Future Combat System



Smart Building Environment

Interconnection, Interoperation, Integration & Scaling Challenges

Low End

High End

Ubiquitous embedded devices

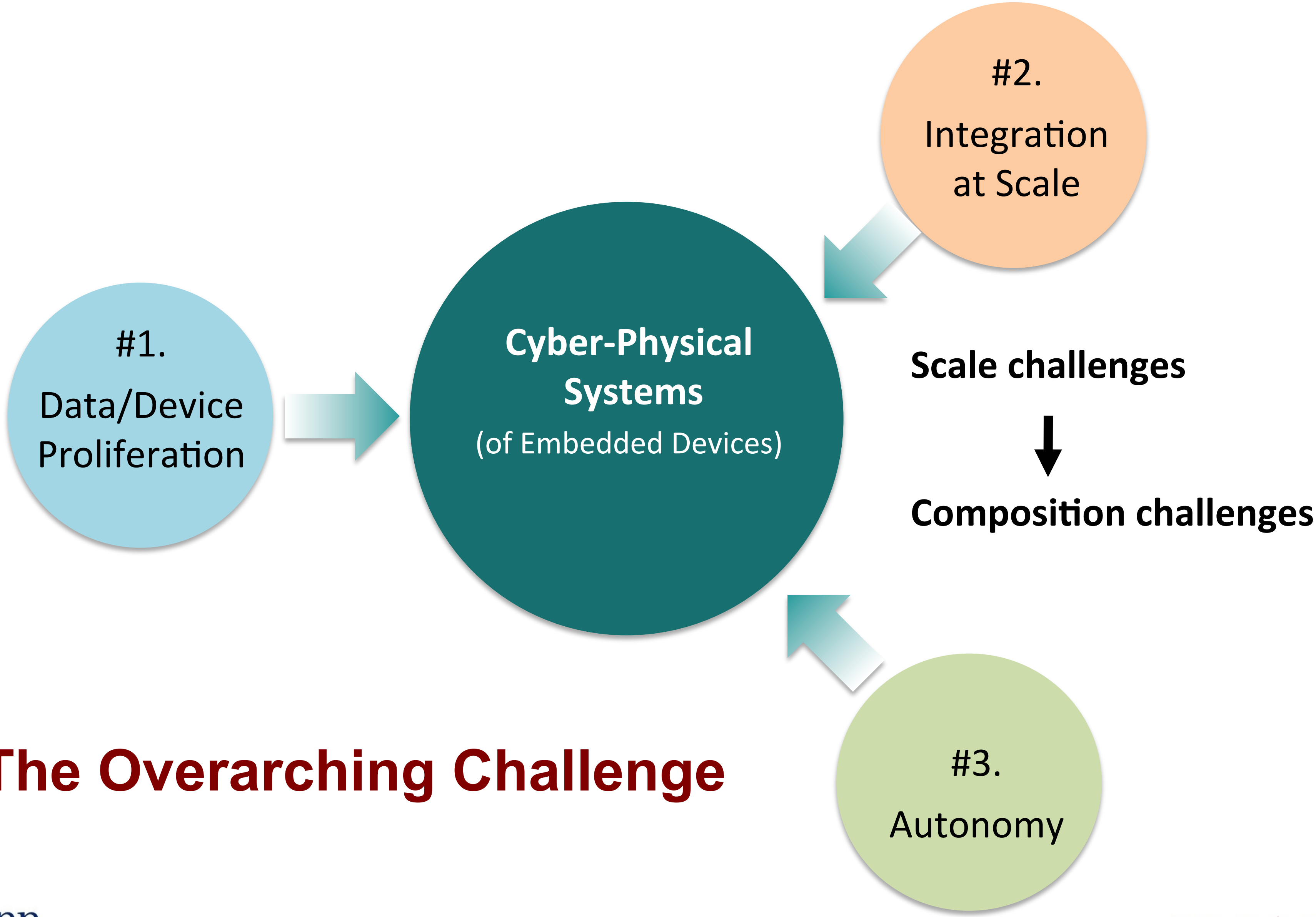
- Large-scale networked embedded systems
- Seamless integration with a physical environment

Complex systems with global integration

- Global Information Grid
- Smart Building Environment

Trend 3: Closing the Loop

Confluence of Trends



#1.

Data/Device Proliferation

#2.

Integration at Scale

#3.

Autonomy

Cyber-Physical Systems

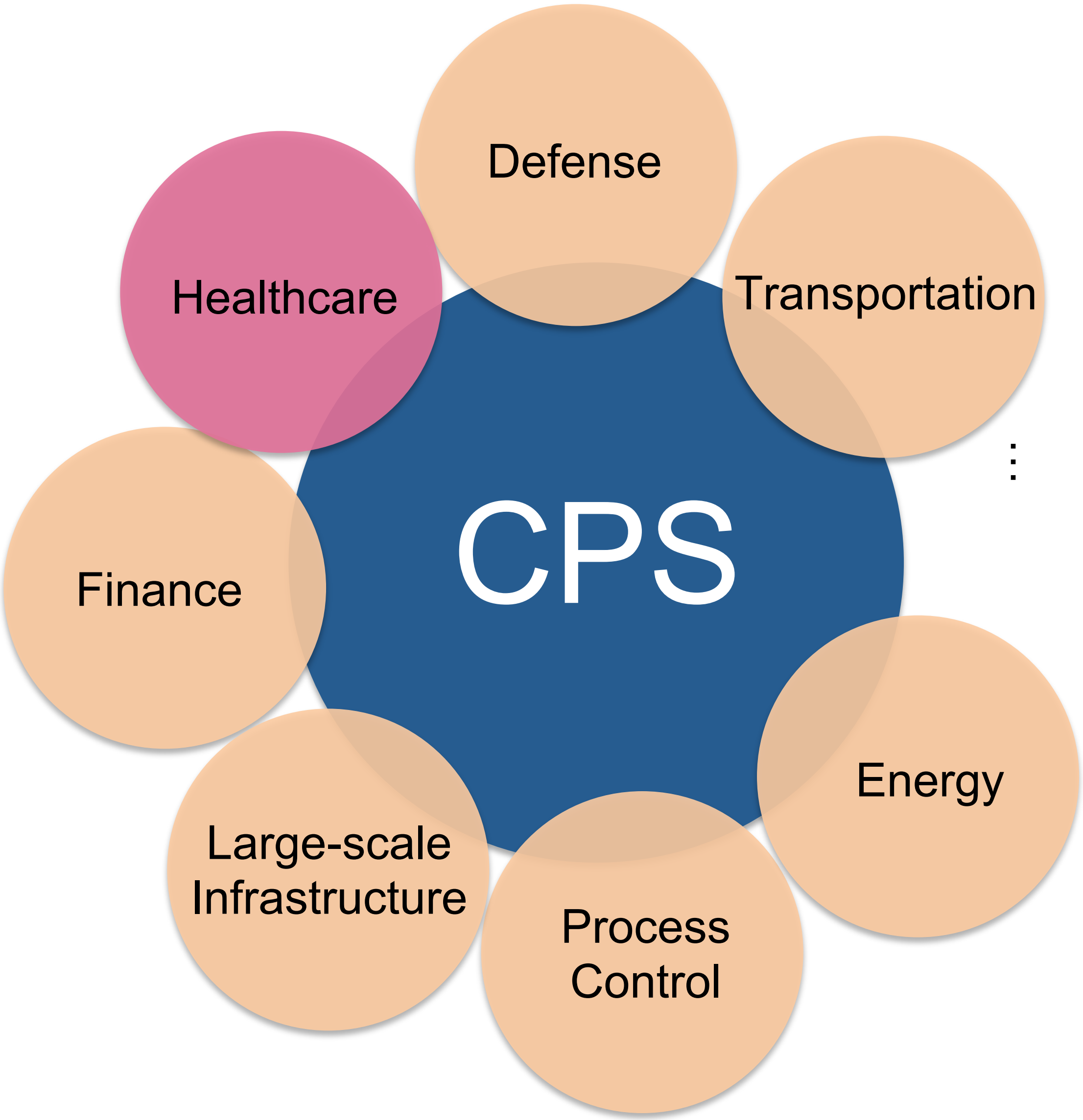
(of Embedded Devices)

Scale challenges

↓

Composition challenges

The Overarching Challenge



CPS

Defense

Healthcare

Transportation

⋮

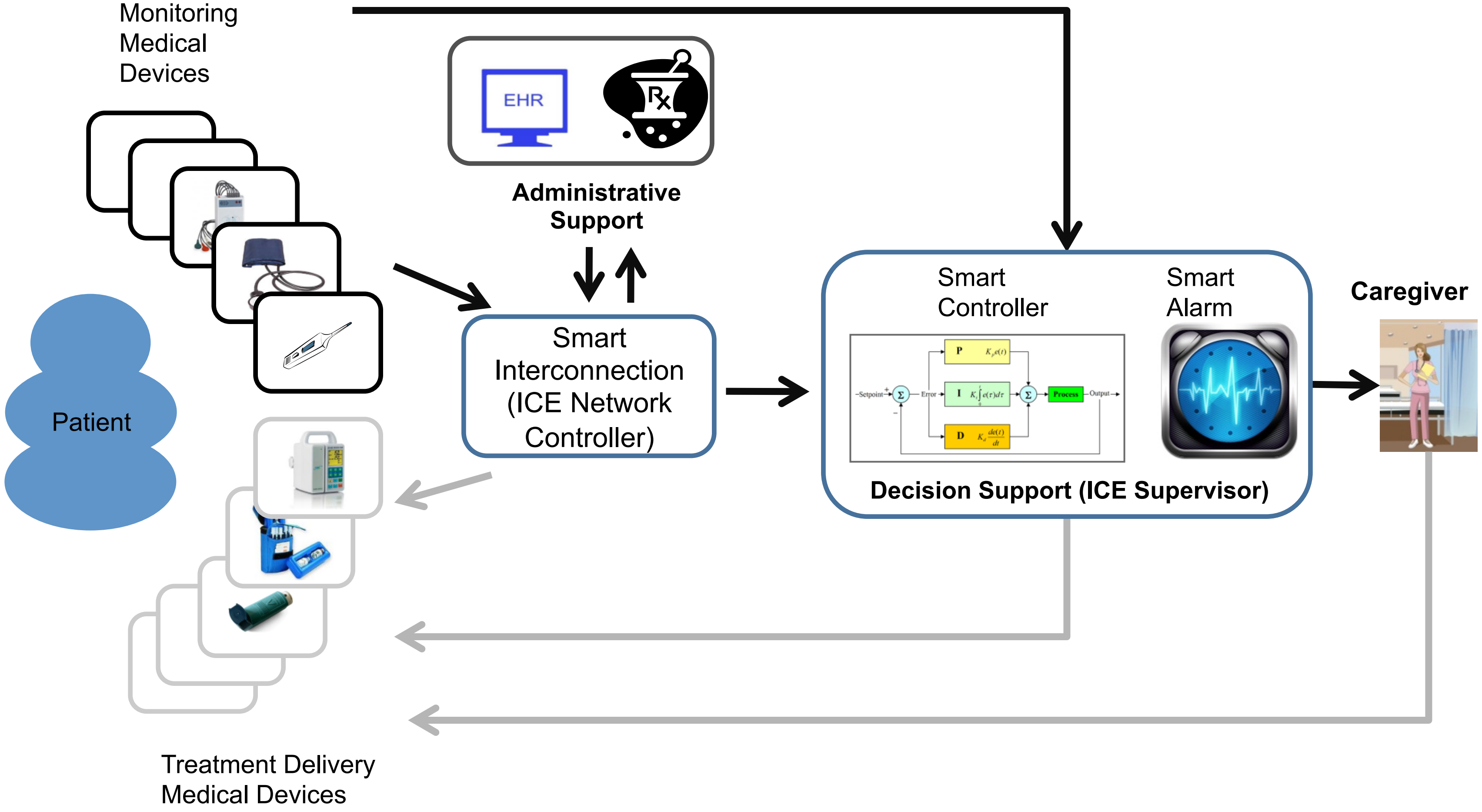
Finance

Energy

Large-scale Infrastructure

Process Control

Overall Structure of MCPS



Monitoring Medical Devices

Administrative Support

Smart Interconnection (ICE Network Controller)

Smart Controller

Smart Alarm

Decision Support (ICE Supervisor)

Caregiver

Patient

Treatment Delivery Medical Devices

MDCPS Research in a Nutshell

- Goal: Develop a new development paradigm for the effective design and implementation of MCPS that are *safe*, *secure*, and *reliable*

- Foundations of MCPS development
  - Patient modeling
  - Caregiver modeling
  - Control-theoretic analysis of physiologically closed-loop scenarios
- High-confidence MCPS software development
  - Model-based development
  - Integration framework for MCPS
  - Security for MCPS
  - [Event recording for medical devices]
- MCPS validation and certification
  - Assurance cases for evidence based certification
  - Compositional techniques for MCPS and assurance cases
- Case studies
  - GPCA, Closed-loop PCA, Pacemaker, Neurological decision support, …

Some Software-related Failures

- Therac-25 (1985-1988)
  - Failure to understand software fault tolerance
- Numerous problems with radiation treatment (http://www.nytimes.com/2010/01/24/health/24radiation.html?ref=radiation)
  - Failures in the generation of treatment plans
- Pacemakers (500K recalls during 1990-2000)
- St Jude pacemaker programmers (2006)
  - Incorrect reporting of pacemaker state
- Difibtech external defibrillators (2007)
  - Self-test resets low-battery status
- Baxter’s Colleague Infusion Pumps (2010)
  - Software update triggers buffer overflow, stops pump

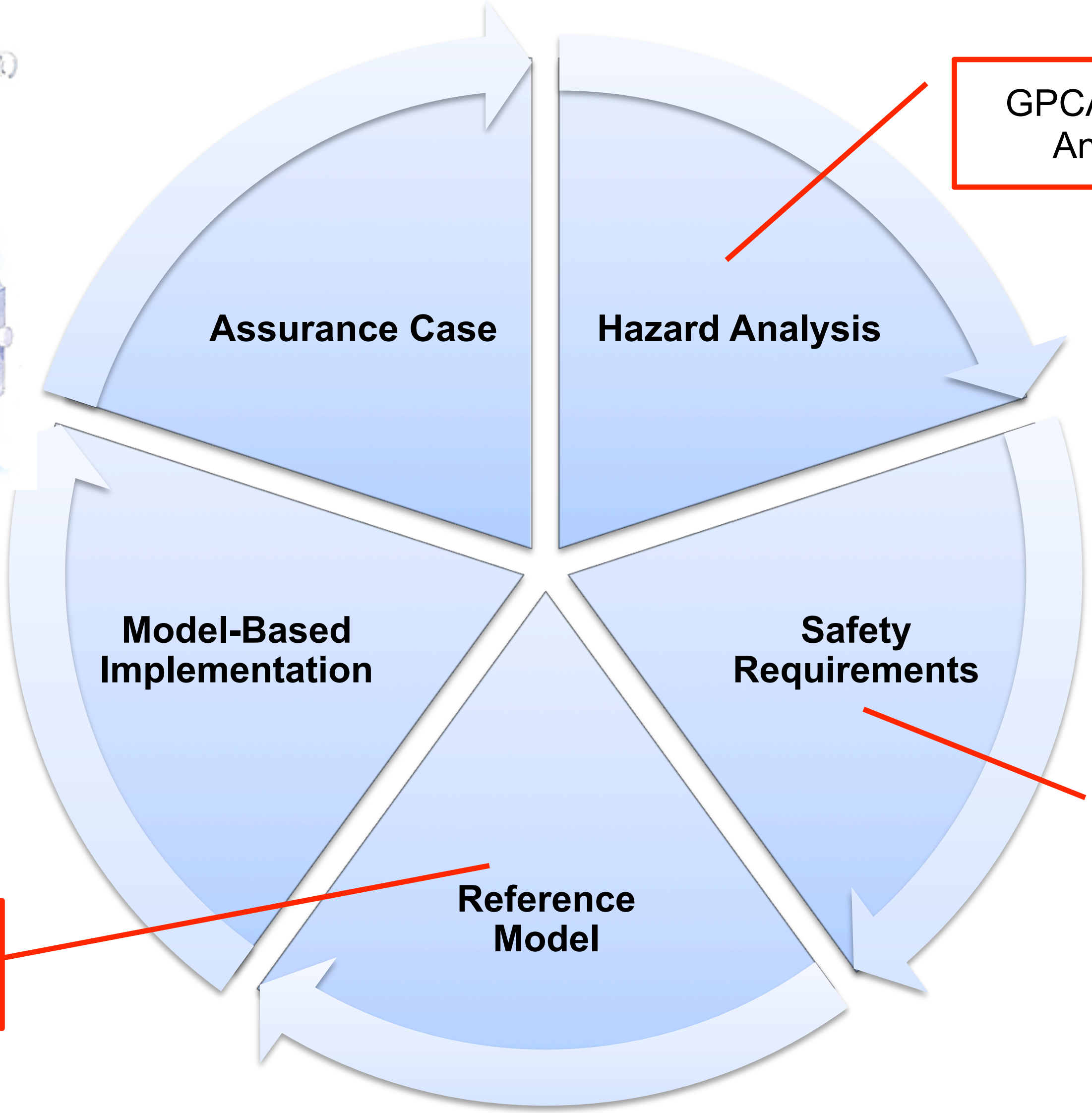
Infusion Pump Safety

- Involved in many clinical accidents
  - During 2005 and 2009, FDA received approximately 56,000 reports of adverse events associated with the use of infusion pumps
  - 1% deaths, 34% serious injuries
  - 87 infusion pump recalls to address safety problems
- The most common types of problems
  - Software Defect
  - User Interface Issues
  - Mechanical or Electrical Failure

U.S. Food and Drug Administration, Center for Devices and Radiological Health. White Paper: Infusion Pump Improvement Initiative, April 2010

Generic PCA (GPCA) Project

PCA Infusion Pump



Assurance Case

Hazard Analysis

Safety Requirements

Reference Model

Model-Based Implementation

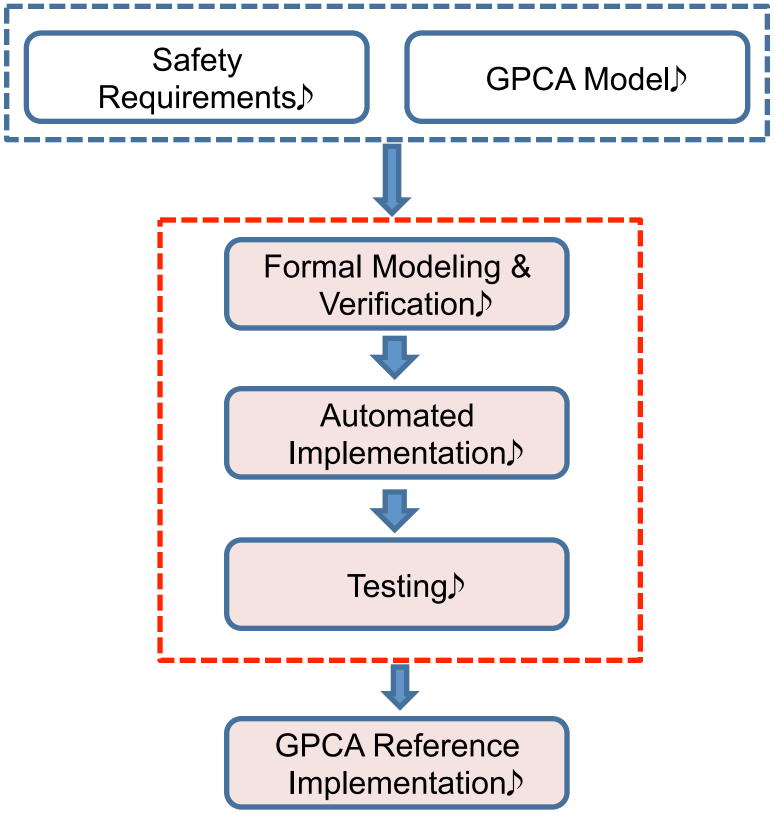
GPCA Hazard Analysis

GPCA Safety Requirements

GPCA Model

GPCA reference implementation

- FDA initiated
  - GPCA Safety Requirements
  - GPCA Model (Simulink/Stateflow)
- Develop a GPCA reference implementation
  - Model-based development
- Provide evidence that the implementation satisfies the safety requirements
  - Safety cases
  - Confidence cases
- All artifacts to be available as open source
  - http://rtg.cis.upenn.edu/gip.php3



Safety Requirements

GPCA Model

Formal Modeling & Verification

Automated Implementation

Testing

GPCA Reference Implementation

Model-Based Development of GPCA Reference Implementation

Quantum related research

- Connectivity, Interoperability, and Compositonality
  - VMD (virtual medical device), VMD app
- Smart Alarms & Decision Support
- Physiological Closed Loop
- Assurance and Certification

Supporting Medical Device Interoperability

Virtual Medical Devices (VMD)

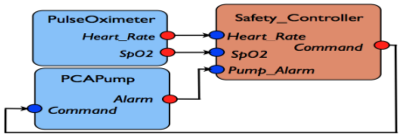
- **MD PnP** (initiative for medical devices interoperability) enables a new kind of medical device, a **Virtual Medical Device (VMD).**
- VMD is a set of medical devices coordinating over a network for clinical scenario. VMD is a virtual system of systems.
- VMD does not physically exist until instantiated at a hospital.
- The Medical Device Coordination Framework (MDCF) is prototype middleware for managing the correct composition of medical devices into VMD.

Device Coordination Algorithm

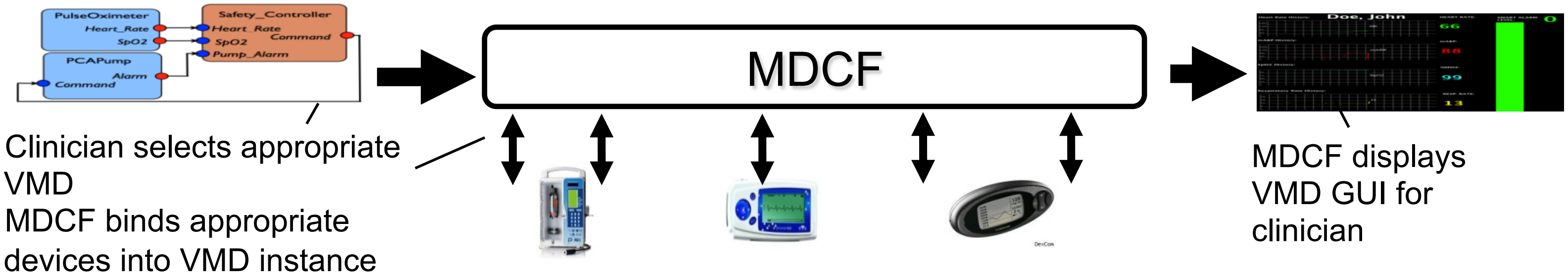
\+

Medical Device Types

=



Virtual Medical Device (VMD)



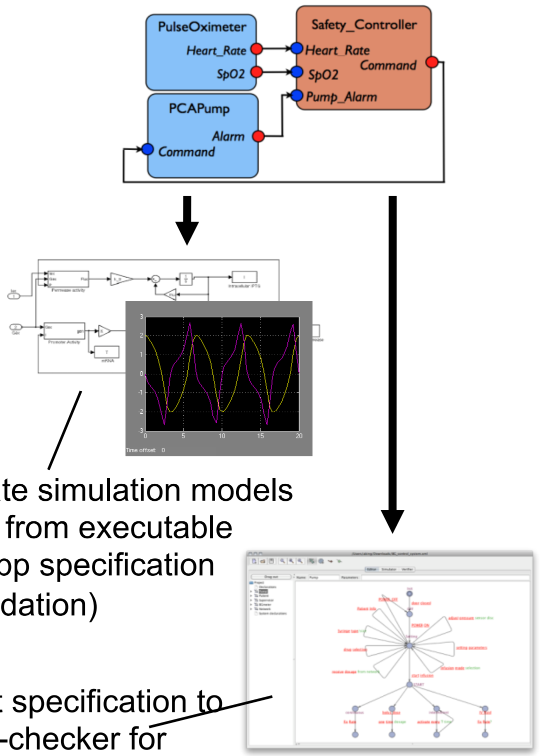
- Clinician selects appropriate VMD
- MDCF binds appropriate devices into VMD instance

MDCF displays VMD GUI for clinician

VMD Research Issues

- **Real-time support**
  - Leverage current hospital networks
- **Non-interference**
  - Assume-guarantee interface
- **Development environment for VMD Apps**
  - Support for programming clinical-algorithms with timing constraints
- **MDCF Platform Implementation**
  - Device connection and configuration protocols
  - VMD setup/tear-down algorithm
  - Guarantee performance specified by VMD App or prevent clinician from unsafely instantiating VMD
- **Safety analysis of the platform**
  - Correctness of the protocols
  - Guarantees of communication

VMD App Validation & Verification



Generate simulation models directly from executable VMD App specification (for validation)

Export specification to model-checker for verification

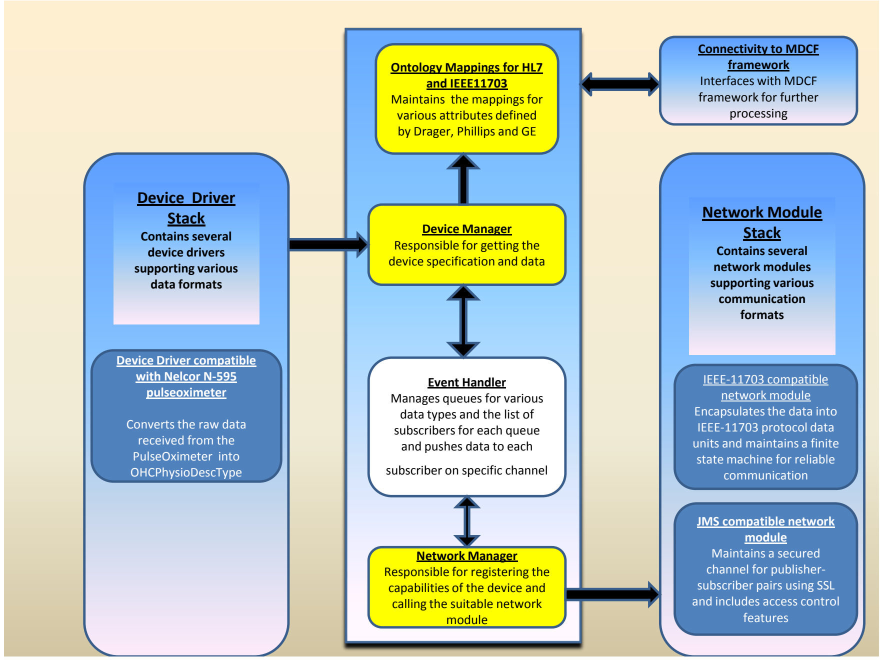
Co-Developed with
NSF CNS-0930647 (PI: John Hatcliff)
Medical Device NIH/NIBIB Quantum Grant (PI: Julian Goldman)

VMD Research Issues

- **Formal VMD requirements and medical device capabilities language**
  - Automatic Device – App compatibility checking by MD PnP platform
    - Ensures correct devices used in any given VMD
    - Reduces scope of standardization efforts to manageable size
      - I.e. standardize the interface language but not the specific "API"
  - Precise VMD development artifact
    - Specs are "executable"
      - Feed into VMD simulation (i.e. testing)
      - Feed into verification (i.e. model checking)
  - Formal semantics
    - May, must, at-least-one of transitions
    - Refinement relations between specification and implementation

Connectivity Support

- Open Health Connector (OHC)
  - Connects legacy devices to modern networks and HIT systems
  - Necessary for MD PnP research
- Open-source, standards-based connectivity
  - Supports 11073 and HL7 messaging
- Customizable
  - Simple patterns and interfaces for implementing new device drivers & network protocols
- Community Support
  - Users contribute back device & network drivers



Device Driver Stack
Contains several device drivers supporting various data formats

Device Driver compatible with Nelcor N-595 pulseoximeter
Converts the raw data received from the PulseOximeter into OHCPhysioDescType

Ontology Mappings for HL7 and IEEE11703
Maintains the mappings for various attributes defined by Drager, Phillips and GE

Device Manager
Responsible for getting the device specification and data

Event Handler
Manages queues for various data types and the list of subscribers for each queue and pushes data to each subscriber on specific channel

Network Manager
Responsible for registering the capabilities of the device and calling the suitable network module

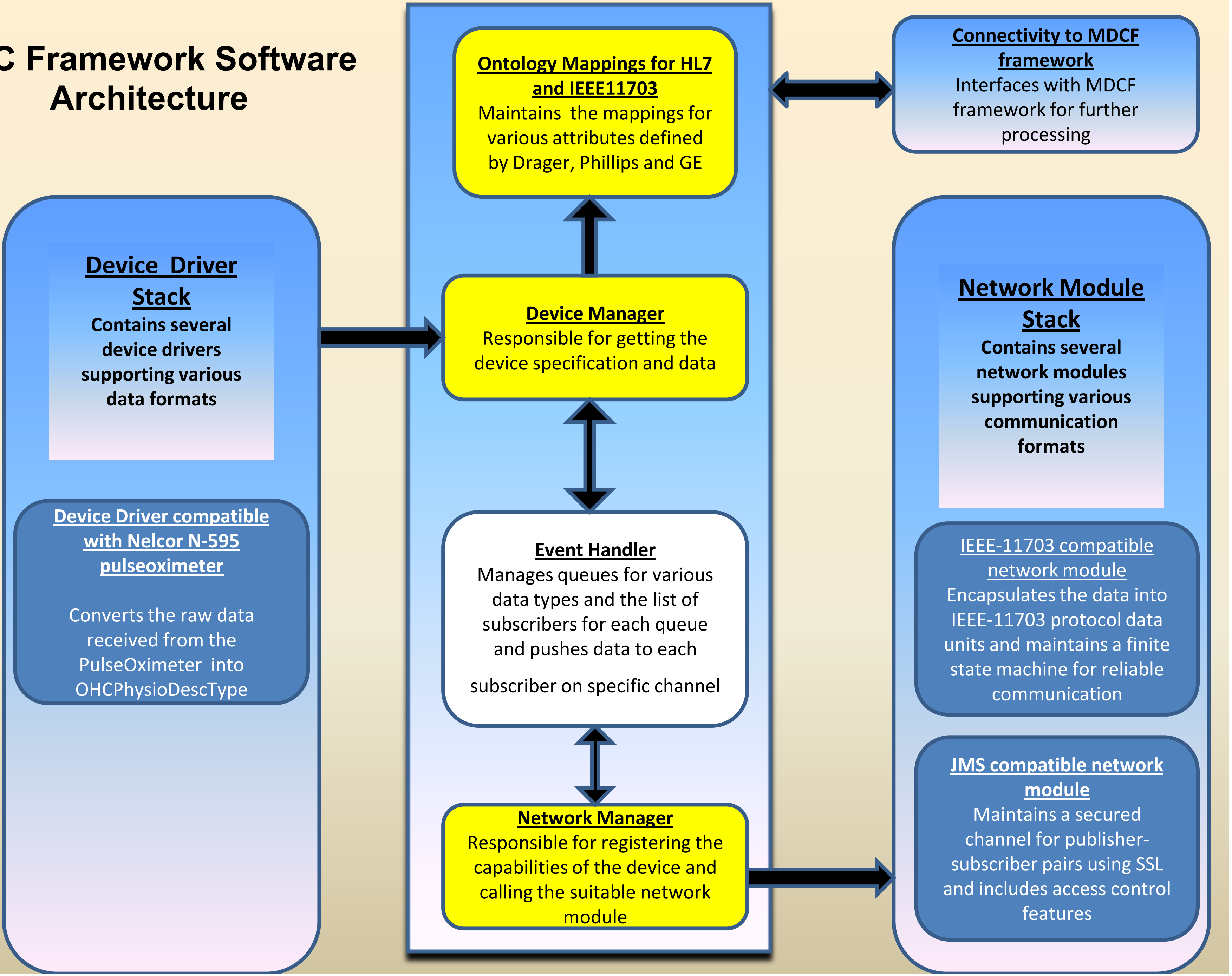
Connectivity to MDCF framework
Interfaces with MDCF framework for further processing

Network Module Stack
Contains several network modules supporting various communication formats

IEEE-11703 compatible network module
Encapsulates the data into IEEE-11703 protocol data units and maintains a finite state machine for reliable communication

JMS compatible network module
Maintains a secured channel for publisher-subscriber pairs using SSL and includes access control features

OHC Framework Software Architecture



Device Driver Stack

Contains several device drivers supporting various data formats

Device Driver compatible with Nelcor N-595 pulseoximeter

Converts the raw data received from the PulseOximeter into OHCPhysioDescType

Ontology Mappings for HL7 and IEEE11703

Maintains the mappings for various attributes defined by Drager, Phillips and GE

Device Manager

Responsible for getting the device specification and data

Event Handler

Manages queues for various data types and the list of subscribers for each queue and pushes data to each subscriber on specific channel

Network Manager

Responsible for registering the capabilities of the device and calling the suitable network module

Connectivity to MDCF framework

Interfaces with MDCF framework for further processing

Network Module Stack

Contains several network modules supporting various communication formats

IEEE-11703 compatible network module

Encapsulates the data into IEEE-11703 protocol data units and maintains a finite state machine for reliable communication

JMS compatible network module

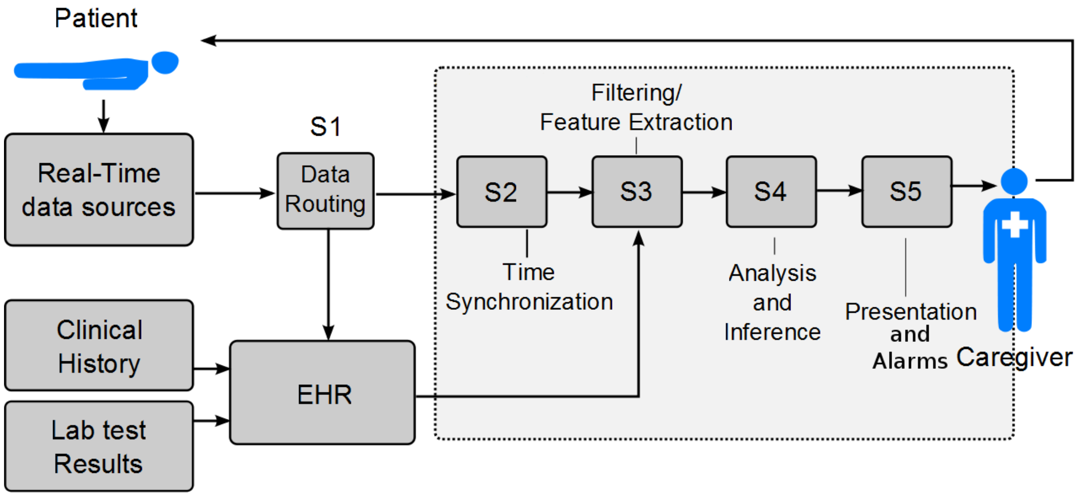
Maintains a secured channel for publisher-subscriber pairs using SSL and includes access control features

Smart Alarms

- 85%-99% of alarms generated in ICUs are false alarms

- **VMD** of multiple devices and central "smart" controller
  - Filter, combine, process, and present real-time medical information
  - Suppress clinically irrelevant alarms
  - Provide summaries of the patient's state and predictions of future trends

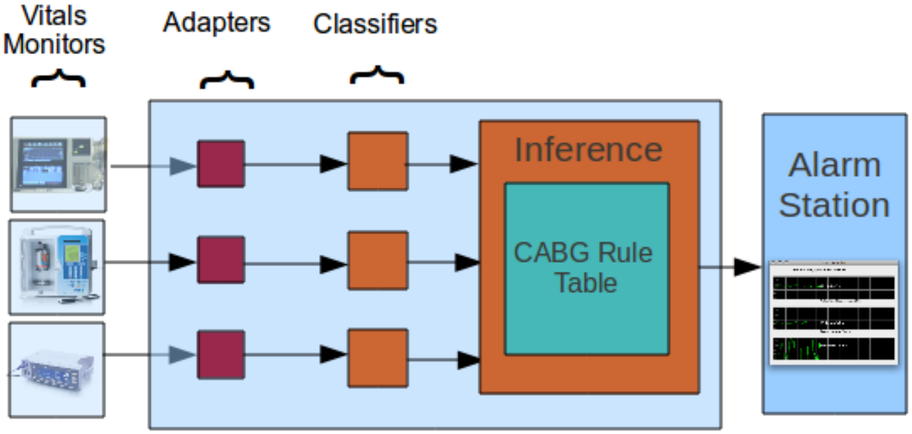
- Benefits
  - Improves patient safety
  - Reduces caregiver workload
  - Facilitates practice of evidence-based medicine



- Challenges
  - Filtering and combining data streams from multiple devices (clock synch?)
  - Developing context-aware patient models
  - Encoding hospital guidelines, extracting experts' models, learning models statistically
  - Presenting data concisely and effectively

Case Study: CABG Smart Alarm

- CABG (Coronary Artery Bypass Graft)
    - Monitoring of post-CABG patients
    - 57% reduction in false alarms
    - No missed true alarms
    - Rule-based, from clinical guidelines and experts
    - Joint work with Margaret Fortino-Mullen, RN



- Non-clinical implementation based on recorded data
- Barriers to real-time deployment

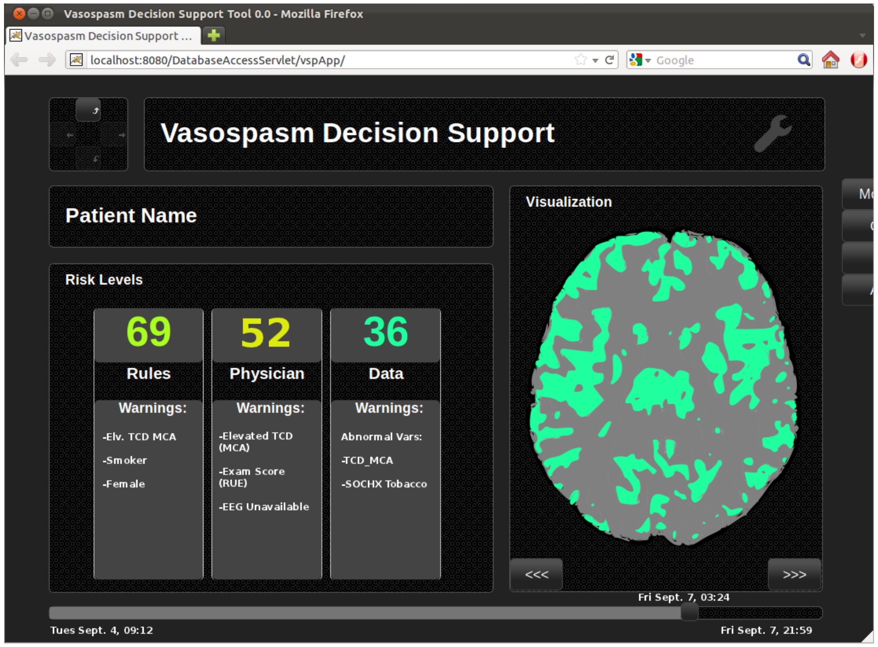
BP	HR	SPO_2	RR	Alarm Level
Normal	Normal	Normal	Normal	0
High	Normal	Normal	Low	1
High	Low	Normal	Normal	2
Very Low	Normal	Normal	High	3
High	High	Low	High	2

Table 1: Small subset of the rule set.

Case Study: Vasospasm Decision Caddy

- Post-brain surgery risk
- Hard to diagnose, deadly if not caught early
- Provide supporting information
  - Context for alarms
  - Give clinicians access to data
  - 15 days of data
- 3-pronged approach
  - Guideline driven
  - Physician driven
  - Data driven
- Current deployment barriers
  - Few real-time data stream feeds
  - No interfacing of streams to the systems
- Joint work with Soojin Park, MD

- Analyze data in new ways
  - New device sources
  - Trending
  - Waveform analysis
  - Clinician provide data
  - Interpolate missing data



localhost:8080/DatabaseAccessServlet/vspApp/

Vasospasm Decision Support

Patient Name

Risk Levels

69	52	36
Rules	Physician	Data
Warnings:	Warnings:	Warnings:
-Elv. TCD MCA -Smoker -Female	-Elevated TCD (MCA) -Exam Score (RUE) -EEG Unavailable	Abnormal Vars: -TCD_MCA -SOCHX Tobacco

Fri Sept. 7, 03:24

Tues Sept. 4, 09:12

Fri Sept. 7, 21:59

Technology Gaps in Smart Alarms/CDS

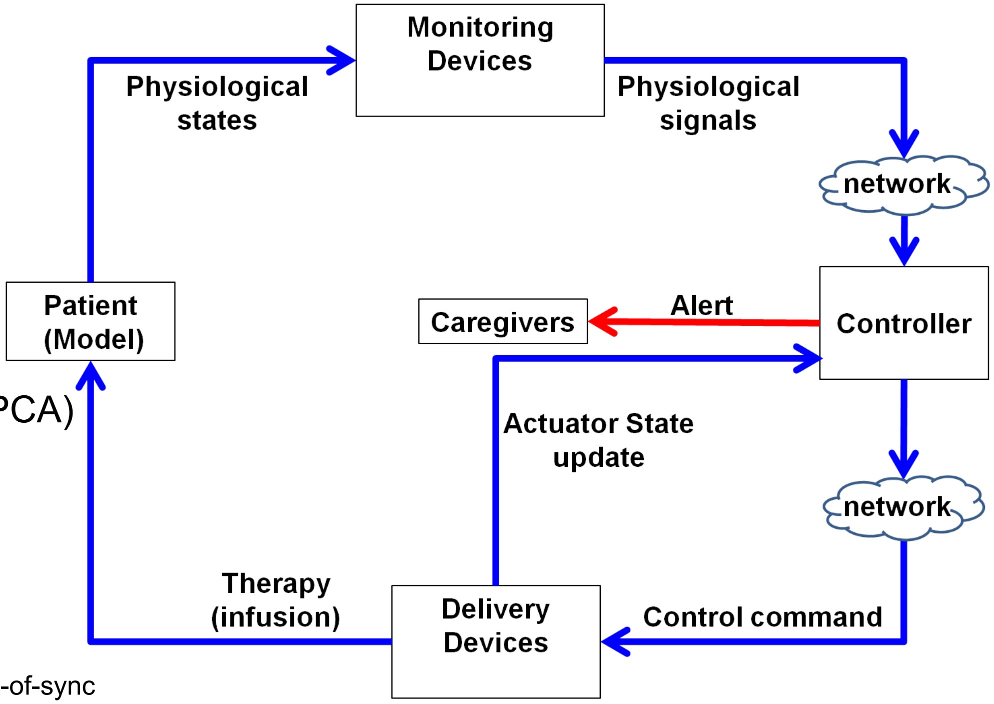
1. Integrating multiple streams of clinical data
2. Poor clock synchronization leads to timing uncertainty, making sensor fusion difficult
3. Safety analysis of Smart Alarms/CDS

4. Translating caregivers' needs into engineering requirements is difficult
   - No "gold standard" for clinical alarms
   - Effective presentation of CDS recommendations

- Interoperability platforms such as ICE standard needed for 1, 2, 3

Physiological Closed-Loop Systems

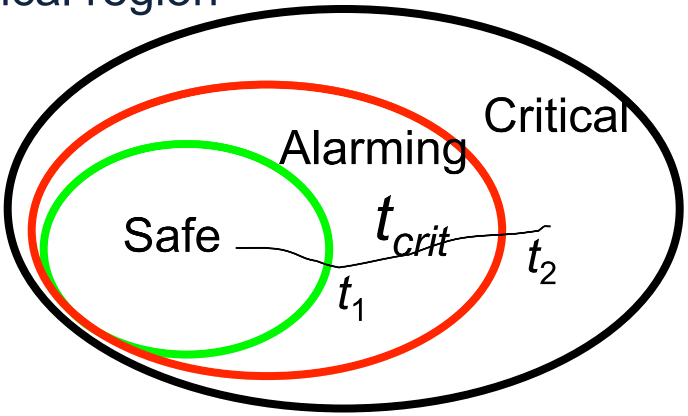
- Benefits
  - Improved patient safety
  - Improved clinical outcomes
  - Reduced deployment cost
    - Networking existing medical devices
- Clinical Use Cases
  - Closed-loop Patient Controlled Analgesia (PCA)
  - Closed-loop Blood Glucose (BG) control
  - Ventilator weaning procedure
- Challenges
  - Hazard identification and mitigation
    - Network packet delay/drop, sensor disconnection, out-of-sync between controllers and devices
  - Verification and Validation
    - Proving safety properties at the model level
    - Validating physiological models with clinical data
- QUANTUM gap
  - Difficult to implement now due to lack of medical device interoperability



PCA Closed-loop System

- Quantum use case
- Goal: improve the safety of PCA
- Approach:
  - Detect respiratory disturbance
  - Provide a safety interlock by stopping the pump
  - Activate nurse call
- Challenges:
  - Patient modeling, large parameter variation
  - New hazards due to network failures
  - Parametric design improves safety but reduces effectiveness

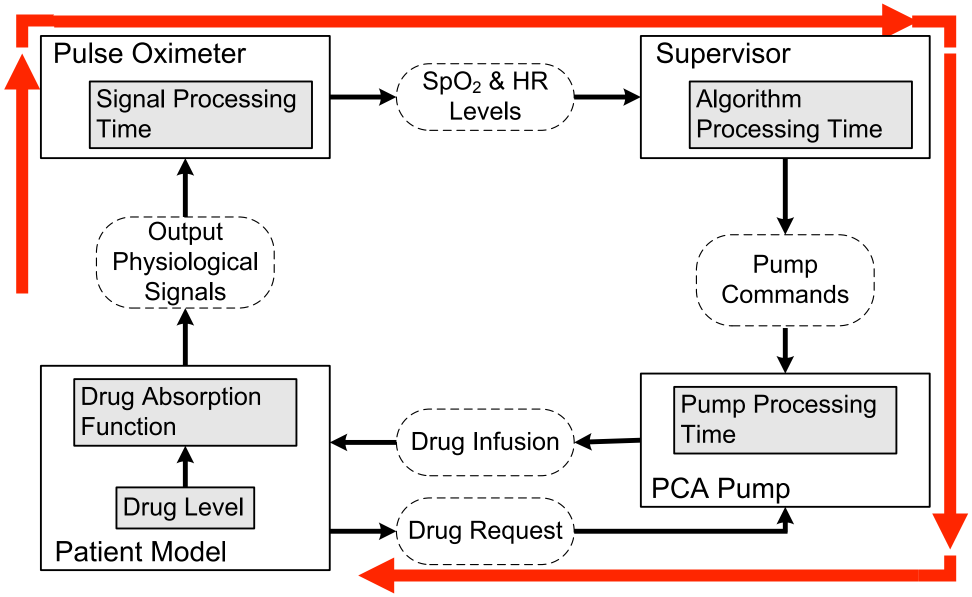
- Safety analysis by formal verification
  - The pump is stopped if patient enters alarming region
  - The patient can not enter the critical region



- Open-loop stability mitigates network hazards
  - Instead of start/stop, allow pump to run for a fixed time

Key Safety Property of Closed-Loop PCA

Pump stops in time if total delay <= t_{crit}



Total delay is the sum of:

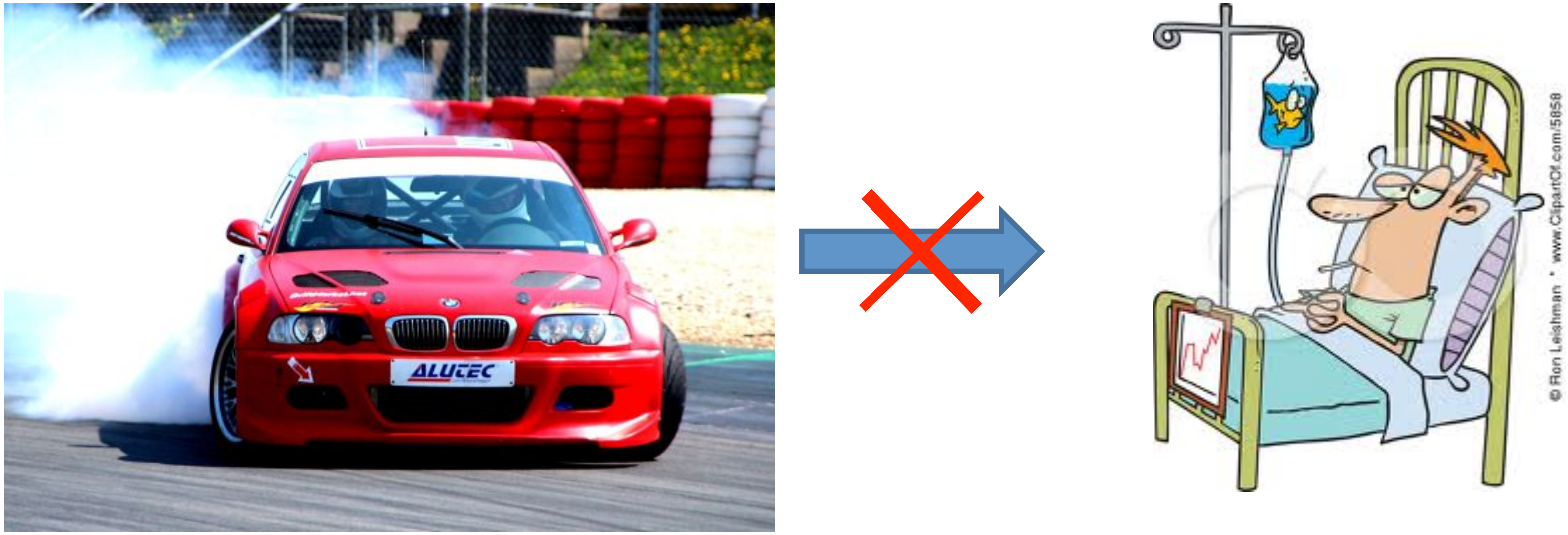
- tPOdel: worst case delay from PO (1s)
- tnet: worst case delay from network (0.5s)
- tSup: worst case delay from Supervisor (0.2s)
- tPump: worst case delay from pump (0.1s)
- tP2PO: worst case latency for pump to stop (2s)
- tcrit: shortest time the patient can spend in the alarming region before going critical

BG Closed-loop System

- Background
  - Glycemic control is important for diabetics and ICU patients
  - Current control guidelines are not adaptive to individual changes and can result in unsafe BG
- Goal:
  - Improve BG control: more in-target time, less variability
  - Minimize hypoglycemia incidents
- Approach:
  - Design controllers on patient models and software simulators
  - At runtime, automatically compute optimal insulin dose and alert caregivers to possible unsafe BG
- Challenges:
  - Patient modeling
    - Not enough data to monitor all physiological states
    - Some factors (e.g., stress, physical activity) are hard to model
  - Sensor measurement errors
  - Actuation (infusion) delays
  - Meal disturbances
- Safety vs. effectiveness
  - Over-aggressive safety algorithm may trigger a lot of hyperglycemia
  - Control design must address this trade-off

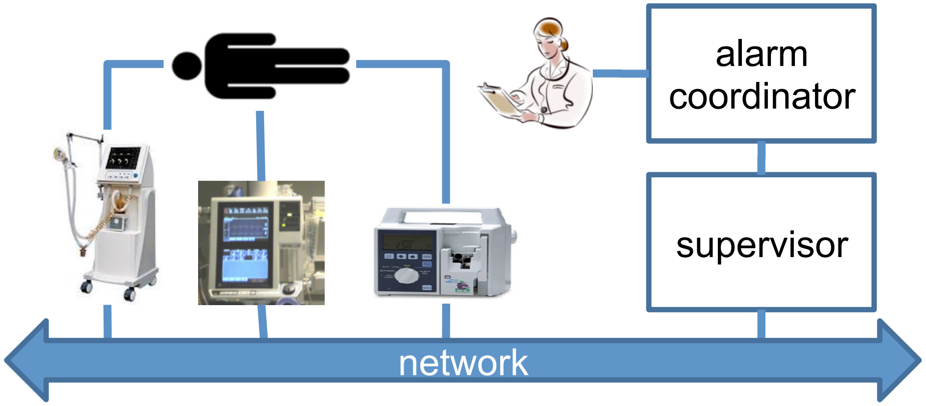
Safe Adaptive Exploration

- Adaptive control often involves learning the parameters by feeding in extreme inputs
  - Example: aggressively turning a car
- Not safe for patient-in-the-loop systems
- Open issue: adaptive exploration with safety constraints



Regulatory Approval of MCPS

- Current approach to certification:
  - Consider every configuration separately



alarm coordinator

supervisor

network

- Cannot be used for MCPS assembled at bedside
  - Multiple devices in the same category
  - Variation in clinical scenarios

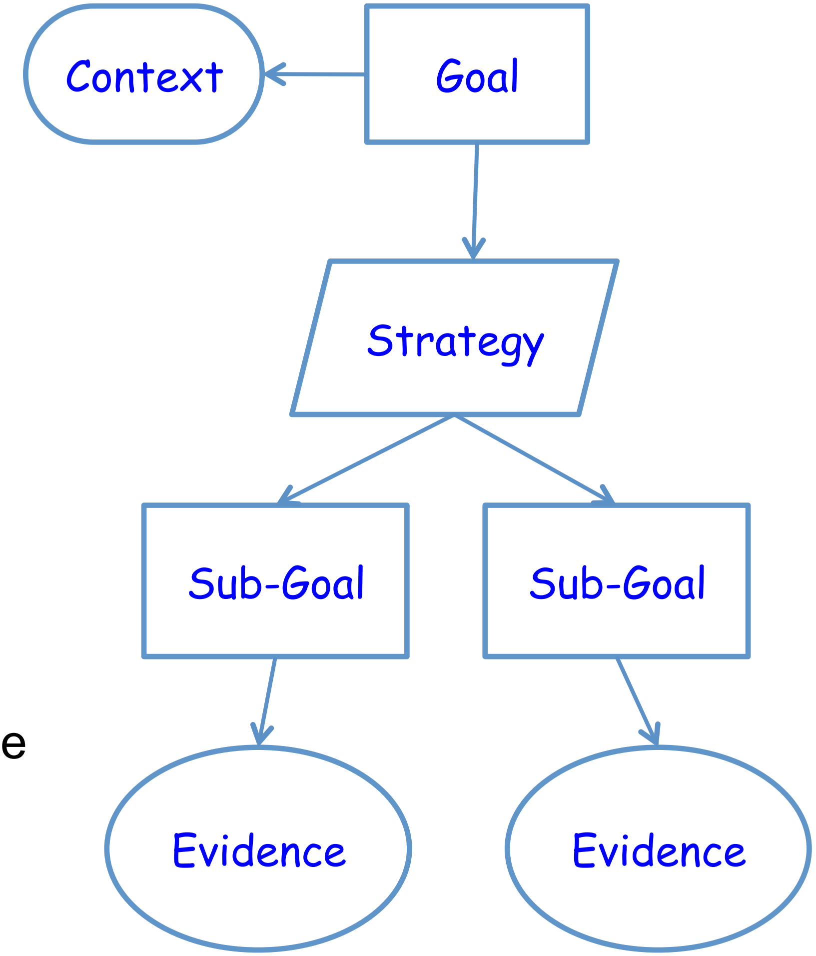
Modular Certification

- An MCPS instance is built to implement a clinical scenario
- Key idea:
  - Treat clinical scenarios as virtual medical devices
- Replace approval of MCPS instances with
  - Certify the scenario
    - Assuming fixed interfaces to constituent devices
  - Certify the interoperability platform
  - Certify devices w.r.t. interfaces

Joint work with J.M. Goldman, J. Hatcliff, A. King, O. Sokolsky, and many others

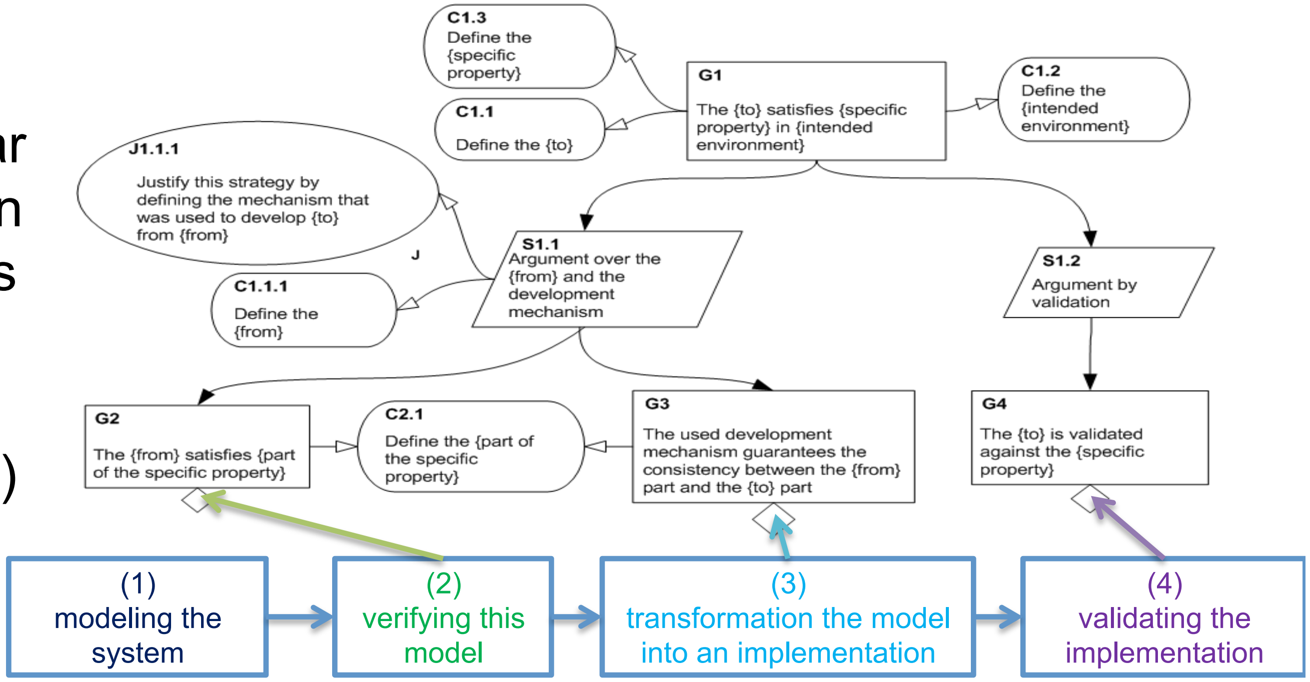
Assurance Cases

- Regulatory Challenge: evidence-based certification
- To construct an assurance case we need to:
  - make an explicit set of claims about the system
  - produce the supporting evidence
  - provide a set of arguments that link the claims to the evidence
  - make clear the assumptions and judgments underlying the arguments
- Challenges and on-going research:
  - Effective ways of constructing assurance cases
  - Evaluation strategies for regulators
  - Certification of interoperating medical devices without N**2 problem



Safety Case Pattern – MDD

- Many devices are developed by similar methods and rely on similar safety claims
- Define a pattern for model-driven development (MDD) approaches



C1.3 Define the {specific property}

C1.1 Define the {to}

G1 The {to} satisfies {specific property} in {intended environment}

C1.2 Define the {intended environment}

J1.1.1 Justify this strategy by defining the mechanism that was used to develop {to} from {from}

J

C1.1.1 Define the {from}

S1.1 Argument over the {from} and the development mechanism

S1.2 Argument by validation

G2 The {from} satisfies {part of the specific property}

C2.1 Define the {part of the specific property}

G3 The used development mechanism guarantees the consistency between the {from} part and the {to} part

G4 The {to} is validated against the {specific property}

(1) modeling the system

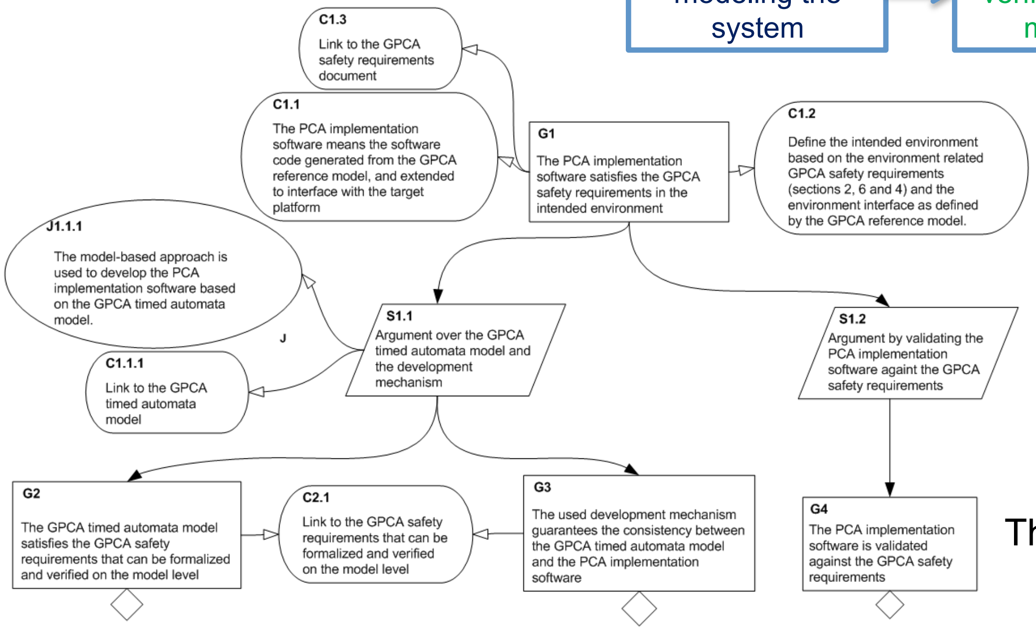
(2) verifying this model

(3) transformation the model into an implementation

(4) validating the implementation

MDD pattern

Instantiation for the PCA safety case



C1.3 Link to the GPCA safety requirements document

C1.1 The PCA implementation software means the software code generated from the GPCA reference model, and extended to interface with the target platform

G1 The PCA implementation software satisfies the GPCA safety requirements in the intended environment

C1.2 Define the intended environment based on the environment related GPCA safety requirements (sections 2, 6 and 4) and the environment interface as defined by the GPCA reference model.

J1.1.1 The model-based approach is used to develop the PCA implementation software based on the GPCA timed automata model.

J

C1.1.1 Link to the GPCA timed automata model

S1.1 Argument over the GPCA timed automata model and the development mechanism

S1.2 Argument by validating the PCA implementation software againt the GPCA safety requirements

G2 The GPCA timed automata model satisfies the GPCA safety requirements that can be formalized and verified on the model level

C2.1 Link to the GPCA safety requirements that can be formalized and verified on the model level

G3 The used development mechanism guarantees the consistency between the GPCA timed automata model and the PCA implementation software

G4 The PCA implementation software is validated against the GPCA safety requirements

The PCA Safety Case – Instance of the MDD pattern

Team members

- Penn, SEAS
  - Insup Lee (PI)
  - Rajeev Alur
  - Rahul Mangharam
  - George Pappas
  - Rita Powell
  - Oleg Sokolsky
- Penn, UPHS/SoM
  - William Hanson, III, MD
  - Margaret Mullen-Fortino, RN
  - Soojin Park, MD
  - Victoria Rich, RN
- Penn, Sociology, SAS
  - Ross Koppel
- MGH/CIMIT
  - Julian Goldman, MD
- Minnesota
  - Mats Heimdahl
  - Nicholas Hopper
  - Yongdae Kim
  - Michael Whalen
- Waterloo
  - Sebastian Fischmeister
- Collaborators
  - John Hatcliff, KSU
  - Paul Jones, FDA
  - Sandy Weininger, FDA
  - Zhang Yi, FDA

CPS: Large: Assuring the Safety, Security and Reliability of Medical Device Cyber Physical Systems (NSF CNS-1035715)

Affiliated Project:

- Medical Device NIH/NIBIB Quantum Grant: Development of a Prototype Healthcare Intranet for Improved Health Outcomes (PI: Julian Goldman)

THANK YOU!