Challenges in Certification for Medical Cyber-Physical Systems

Insup Lee and Oleg Sokolsky
PRECISE Center
School of Engineering and Applied Science
University of Pennsylvania

S5 Workshop
June 14, 2012
Miniaturization
• Implantable devices
• Ingestible sensors

Interoperation
• Executable clinical scenarios
• Safety interlocks

Teleoperation
• Tele-ICU
• Robotic surgery

Autonomy
• Smart alarms
• Context-sensitive decision support
• Physiological closed loop control
Problem/Motivation

• Complexity of medical device systems is rapidly increasing
  – Traditional approaches to regulatory approval are quickly becoming inadequate
  – Existing certification approaches may soon become an obstacle to clinical innovation

• Interconnected medical devices present a special challenge for certification
Outline

- Motivation: multi-device clinical scenarios
- Medical device interoperability
- Virtual Medical Device (VMD) app
- Integrated system certification
  - System-based certification
  - Component-based certification
- Other CPS domains
Certification

• In the U.S., FDA approves medical devices for specific use
  – Safety and effectiveness are assessed
  – Evaluation is process-based: ISO 9001 (quality management) and ISO 14971 (risk management)
  – Hazard analysis is key to approval
  – FDA’s 510(k) requires “substantially equivalent” to devices on the market

• Certification of interconnectable MCPS
  – Currently, each collection of interconnected devices is a new medical device to be approved.
Use Case: X-Ray / Ventilator

“With the advent of sophisticated anesthesia machines incorporating comprehensive monitoring, it is easy to forget that serious anesthesia mishaps still can and do occur.”
APSF Newsletter Winter 2005
Another Use Case: PCA Monitoring

- Patients are commonly given patient-controlled analgesics after surgery
- Crucial to care, but numerous issues related to safety

A 49-year old woman underwent an uneventful operation (total abdominal hysterectomy and bilateral salpingo-oophorectomy). Postoperatively, the patient complained of severe pain and received intravenous morphine sulfate in small increments. She began receiving a continuous infusion of morphine via a patient controlled analgesia (PCA) pump. A few hours after leaving the PACU [post anesthesia care unit] and arriving on the flow, she was found pale with shallow breathing, a faint pulse, and pinpoint pupils. The nursing staff called a "code", and the patient was resuscitated and transferred to the intensive care unit on a respirator. Based on family wishes, life support was withdrawn and the patient died. Review of the case implicated a PCA overdose. Delayed detection of respiratory compromise in patients undergoing PCA therapy is not uncommon because monitoring of respiratory status has been confounded by excessive nuisance alarms.
Simple Closed Loop Control

Motivating Clinical Problem: PCA Overdose

“A particularly attractive feature may be the ability to automatically terminate or reduce PCA (or PCEA) infusions when monitoring technology suggests the presence of opioid-induced respiratory depression. To facilitate such capabilities, we strongly endorse the efforts to develop international standards for device interoperability and device-device communication.

It is critical that any monitoring system be linked to a reliable process to summon a competent health care professional to the patient's bedside in a timely manner.”

[Goldman]
PCA Closed-loop System

• Goal: Improve the safety of PCA uses

• Approach: Integrate monitors with an intelligent “controller” to:
  – Detect respiratory disturbance
  – Safety lock over infusion
  – Activate nurse-call
What do X-Ray/Ventilator and closed loop PCA interlocks have in common?

They do not exist!

Why not?
Medical-Device Plug-and-Play

Characteristics

- Medical devices gaining communication capabilities
- Devices still operate independently
- Standardized interaction between devices nonexistent
- Full benefit of communication capabilities not being realized

Advantages

- Improve Patient safety
- Safety interlocks
- Complete, accurate medical records
- Reduce errors
- Context awareness
- Rapid deployment

MD PnP: Interoperable medical devices based on plug-n-play!
Vendor neutrality based on open medical device interfaces
www.mdpnp.org
Supporting Medical Device Interoperability

Subsequent standards are intended to provide specific functional and interfacing requirements for components.

The ICE architecture standard is the focal point for FDA’s evaluation of MAP concepts in future medical systems. A key element of this evaluation is moving from regulation of “systems as a whole” to component-wise regulation.
Virtual Medical Devices (VMD)

- **MD PnP** enables the concept of **Virtual Medical Devices**: A set of medical devices coordinating over a network for clinical scenario.

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.

- Clinician selects appropriate VMD
- MDCF binds appropriate devices into VMD instance
Development & Assembly of VMD

• Identify and define clinical scenario
  – Component medical devices
  – Information flows and their parameters
  – System of systems to support specific clinical scenario

• Model and verify safety properties
  – Identify hazards and safety properties
  – Assume VMD platform

• Implement VMD app
  – Modes: Initialization, Operation, Shut-down
  – Runtime assurance needed during each mode
  – Middleware platform to support VMD app
VMD Research Issues

- Real-time support
- Non-interference
- 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 John Hatcliff and Julian Goldman
Safety Analysis of VMD Apps

- Closed-loop PCA as an example

Pharmacokinetics:

Hazard analysis of the scenario:
- New hazards due to network
- Open-loop stability to mitigate new hazards

Key safety properties proved
Certification of VMD App

• Safety analysis of the VMD model relies on assumptions about
  – Devices that comprise the VMD
  – Interoperability infrastructure

• Current regulatory approach:
  – Certify each instantiation of VMD app
    • fixed medical devices, network, middleware, etc.

• Alternative approach:
  – Certify VMD app based on abstract interfaces
  – Certify devices on interface satisfaction
  – System can use any certified component
Traditional safety critical systems...

Aerospace

Nuclear

Automotive
System Integration

In other safety critical domains, there is typically a prime contractor that is responsible for integration and system-level verification and validation.

- Integration is performed *before* deployment with full knowledge and behavior of components being integrated.
- Integrator has expert-level technical knowledge of components & system behavior.
- Responsible for overall system:
  - Verification & Validation
  - Safety arguments
  - Certification

787 Final Assembly Integrator - The Boeing Company

As Prime Contractor/Integrator for the final assembly of the composite 787 Dreamliner in Everett, WA, Boeing is the prime contractor and weapon system integrator for the EA-18G Growler and leads the industry team, which includes Northrop Grumman as principal subcontractor and airborne electronic attack subsystem integrator.
In other safety critical domains, there is typically a prime contractor that is responsible for integration and system-level verification and validation.

End to end process managed by prime contractor.

System integration and V & V is done before system is delivered to the customer.
VMD Development & Assembly

Medical Device Manufacturer

VMD Platform Manufacturer

VMD App Developer

ConOps

Requirements

Design

Impl / V & V

App Execution
(dynamic formation of MAP constituted device)

Performed by runtime environment of VMD

Performed by clinical staff

VMD Instance Assembly

Deployment

Market
VMD Characteristics

In other safety critical domains, there is a typically a prime contractor that is responsible for integration and system-level verification and validation.

- Integration is performed before deployment with full knowledge and behavior of components being integrated
- Integrator has expert-level technical knowledge of components & system behavior
- Responsible for overall system
  - Verification & Validation
  - Safety arguments
  - Certification

With VMDs, there is no prime contractor that is responsible for integration and system-level verification and validation.

- Assembly is performed after deployment
- Assembler (hospital staff) does not have expert-level technical knowledge of components & system behavior
- App developer is responsible for overall system safety arguments
- Platform services (compatibility checks) assist in determining at app launch time if platform and attached devices satisfy requirements of app
- The app’s directions for assembly of the platform constitute device are stated only in terms of properties/capabilities that are exposed on the interfaces of the platform and devices.
Current Regulatory Approach

Current regulation of integrated systems (e.g., central station monitors) requires "pair-wise" clearance: whenever a new type of device is added to the monitoring platform, the entire infrastructure must be re-cleared.

In current regulatory approach, adding a new type of device (e.g., Z) typically causes the entire system to be re-submitted for regulatory clearance.
The Bottom Line

• Why traditional approach won’t scale
  – Medical devices need to work as stand alone as well as an integrated system

• Certify VMD app based on abstract interfaces
  – System can use components (including medical devices) that satisfy their specs
Pairwise Approval / Certification

Example “interoperable” device ecosystem 3 different (model/manufacturer) blood sugar sensors, 3 different (model/manufacturer) insulin pumps:

Each sensor must be approved or certified for use with each pump and vice versa. This is burdensome for manufacturers and regulators.
Interface-based approval / certification

Example “interoperable” device ecosystem 3 different (model/manufacturer) blood sugar sensors, 3 different (model/manufacturer) insulin pumps:

Each sensor (or pump) only needs certification or approval w.r.t. the interface spec. Additionally, the ecosystem can grow without forcing recertification (or re-approval) of previously analyzed devices.
Safety claim for an app

- Operation is safe, as long as
  - Each device satisfies a set of requirements
  - Communication proceeds according to the model
- Requirements for devices
  - Functional capabilities
  - Alarm reporting
  - Quality of data, etc.
- Communication requirements
  - QoS guarantees
Modular Certification

• Certify VMD platform
• Certify VMD app
• Certify medical device confirm to VMD interface
  – There is a catch: there may be zillions of VMDs
  – Undoubtedly classes of VMDs with similar interfaces will emerge

• Is it a viable regulatory path?
Making Assumptions Explicit

• In order for the claim to be valid, assumptions need to be checked
  – Some can be checked at assembly time
    • E.g., each device has the right capabilities
  – Others have to be checked at run time

• Infrastructure needs to be certified as capable of supporting app safety
  – Device capability checking and enforcement
  – Run-time assurance concepts can be used to enforce safety
    • Monitor device status and communication process
Conclusions

- Traditional integrated system certification approach does not scale to VMD apps
- Can we certify VMD app that can be instantiated at runtime
  - Abstract Certification base on Interface
- What evidence does one need for abstract certification?
  - Show safe, secure, and effective
  - Integrated systems without integrator
- How can we assure that interfaces are met?
  - 3rd party certifier
  - Runtime checking
- Can we use this technique for certified components?
  - Crowd sourcing of safety critical apps
Acknowledgements

• Domain knowledge comes from our collaborators at the Upenn Hospital and MD PnP program at CIMIT
  – Margaret Mullen-Fortino, RN
  – Julian Goldman, MD

• VMD concepts are developed in collaboration with KSU (John Hatcliff) and MD PnP Quantum project

• Generous funding from NSF and NIH

• Some of the slides are courtesy of Julian Goldman, John Hatcliff and Andrew King
Thank You!
Questions?