Open Source Patient-Controlled Analgesic Pump Requirements Documentation

Brian R. Larson, John Hatcliff, Patrice Chalin
Kansas State University, Kansas, USA
{brl,hatcliff,chalin}@ksu.edu

***Abstract*—The dynamic nature of the medical domain is driving a need for continuous innovation and improvement in techniques for developing and assuring medical devices. Unfortunately, research in academia and communication between academics, industrial engineers, and regulatory authorities is hampered by the lack of realistic non-proprietary development artifacts for medical devices. In this paper, we give an overview of a detailed requirements document for a Patient-Controlled Analgesic (PCA) pump developed under the US NSF's Food and Drug Administration (FDA) Scholar-in-Residence (SIR) program. This 60+ page document follows the methodology outlined in the US Federal Aviation Administrations (FAA) Requirements Engineering Management Handbook (REMH) and includes a domain overview, use cases, statements of safety & security requirements, and formal top-level system architectural description. Based on previous experience with release of a requirements document for a cardiac pacemaker that spawned a number of research and pedagogical activities, we believe that the described PCA requirements document can be an important research enabler within the formal methods and software engineering communities.**

I. Introduction

In the ninetieth century, physicians sometimes practiced grave-robbing to obtain subjects for investigation. If only finding suitable subjects for application of software engineering and formal methods in the medical domain were so easy.

There are a number of desirable qualities of case study artifacts for facilitating research and pedagogy in the medical device domain:

- the subject matter must be "real-world" enough to be relevant;
- it must be supported by domain documentation including: appropriate background on device mechanics needed for the targeted physiological monitoring and actuation, relevant human physiology, information on typical clinical contexts including use cases and clinical workflows;
- it must be "big" enough to show methods scale, yet not overwhelm small academic teams,
- it should expose *systems* issues—both software and hardware functionality should be exposed to a degree of specificity needed to support work on techniques for risk management, hazard analysis, and system safety;
- it should include or provide a pathway for execution on actual hardware or realistic simulation so as to enable realistic evaluation of testing, verification, and other quality assurance techniques;
- it should include information sufficient for enabling academic teams to be aware of, and even develop, techniques for addressing regulatory and certification issues.

This paper describes a case-study artifact, a publicly-available requirements document [1], that possesses many of the qualities identified above and provides a pathway for realizing those remaining.

A. Previous Experience with Case Studies that Catalyze Research and Education

While working as an engineer at Boston Scientific, significant efforts by the first author resulted in the release into the public domain of a system specification for a previous generation implantable cardiac pacemaker [2]. The goal of this effort was to catalyze research and education on realistic applications of formal methods and evidence-based certification regimes. Larson advised students at the University of Minnesota and faculty at McMaster University in developing an inexpensive hardware platform for class projects on which pacemaker code could be executed/simulated and guidelines for evaluating solutions submitted in response to verification and certification challenge problems. McMaster University researchers and the Software Certification Consortium (SCC) developed and supported the "Pacemaker Formal Methods Challenge"[1], which led to several special workshops and events that focused on highlighting formal methods. An upcoming Dagstuhl Seminar is dedicated to reporting on past work and facilitating future research related to the pacemaker artifacts. Up to this point, the pacemaker requirements document has been utilized in more than 30 publications and in class projects at a number of universities.

B. Goals of This Work

In this paper, we report on an effort that aims to have a similar catalyzing effect but this time for Patient-Controlled Analgesic (PCA) pumps. We give an overview of a detailed requirements document for a PCA pump developed in consultation with US Food and Drug Administration (FDA) engineers under the auspices of the US National Science Foundation (NSF) Food and Drug Administration's (FDA) Scholar In Residence (SIR) program. There are a several important features of this document:

[1] sqrl.mcmaster.ca/pacemaker.htm

978-1-4673-6282-5/13 © 2013 IEEE

SEHC 2013, San Francisco, CA, USA

- the 60+ page document follows the methodology outlined in the US Federal Aviation Administrations (FAA) Requirements Engineering Management Handbook (REMH) [3];
- it includes a domain overview providing relevant clinical context;
- it provides a collection of normal and exceptional use cases, as well as
- a formal architecture description including both software and hardware components specified using the SAE standard Architecture and Analysis Definition Language (AADL) [4], [5],
- the architecture is organized to provide a distinct *safety architecture*—a separate subsystem designed to monitor for system faults and take appropriate action to mitigate associated hazards and ensure patient safety.

The requirements document is already being utilized in different settings, which we plan to report on in detail elsewhere. Within our own research, the first author has led an effort to supplement the AADL architecture description with formal behavioral specifications written in his BLESS framework [6]. These specifications include contracts on component boundaries written in the BLESS behavioral interface specification language and formal proofs of behavioral conformance to contracts using the BLESS proof tool. We are also using the requirements specification to support research on interoperability, safety, and security for devices that interoperate following the architecture specified in the ASTM Integrated Clinical Environment standard [7]. The requirements document is also supporting collaborative work with the FDA and Underwriters Laboratory on safety standards for interoperable medical devices. Finally, we are contributing to the Software Certification Consortium [8], which is seeking such problems from many safety-critical domains for "mock" certifications including assurance cases arguing for safety and effectiveness from evidence, especially from formal methods.

C. Previous Work

Kansas State's PCA Pump requirements document builds upon University of Pennsylvania's and FDA's Generic Infusion Pump (GIP) project [9]. The GIP project includes a smaller set of requirements, and an initial hazard analysis for the pump. These GIP artifacts were utilized in follow-on work by researchers at UPenn / FDA and elsewhere on the application of verification techniques that tended to emphasize properties that could be checked by real-time model checkers like UPPAAL [10]. Our work aims to further the objectives of the GIP project by expanding on the original requirements document along several dimensions, *e.g.*, by significantly expanding the requirements to address a much broader set of functionality and additional safety requirements, by adding clinical motivation and use case descriptions, by adding formal architectural descriptions, by introducing a safety architecture, and by aligning the document with the methodology suggested in the FAA REMH.

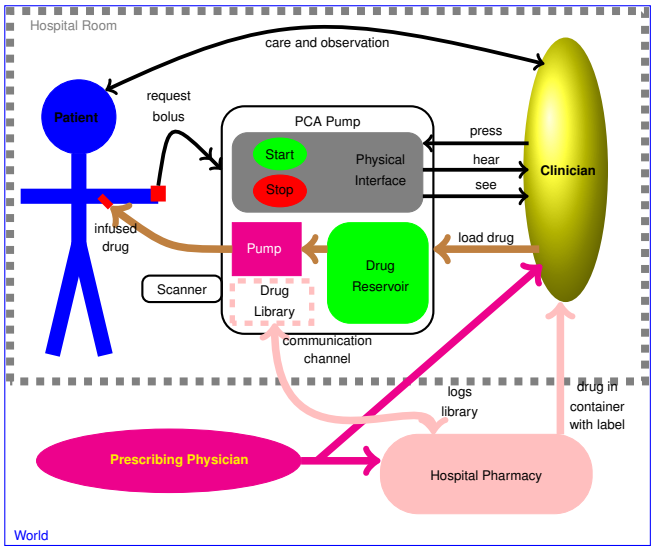

Fig. 1. Independant PCA Pump Use

II. PCA Pump Background

A PCA infusion pump is used to infuse a pain killer. Pain medication is prescribed by a licensed physician, which is dispensed by the hospital's pharmacy. The drug is placed into a vial labeled with the name of the drug, its concentration, the prescription, and the intended patient. A clinician loads the drug into the pump, and attaches it to the patient. The pump infuses a prescribed basal flow rate which may be augmented by a patient-requested bolus or a clinician-requested bolus. This allows additional pain medication in response to patient need within safe limits.

PCA pumps, unfortunately, have been associated with a large number of adverse events [11], [12]. The FDA notes [13] that while PCA pumps (and infusion pumps in general) have allowed for a greater level of control, accuracy, and precision in drug delivery—thereby reducing medication errors and contributing to improvements in patient care—infusion pumps have been associated with persistent safety problems. From 2005 through 2009, 87 infusion pump recalls were conducted by firms to address identified safety problems. Infusion pump problems have been observed across multiple manufacturers and pump types. Through analysis of pump-related adverse event reports and device recalls, FDA has concluded that many of these problems appear to be related to deficiencies in device design and engineering.

Through the *Infusion Pump Improvement Initiative* [13], FDA is taking broad steps to prevent infusion pump problems. Specifically, FDA aims to establish additional requirements for infusion pump manufacturers, proactively facilitate device improvements, and increase user awareness of problems and best engineering practices. As an example of emphasizing best engineering practices, the FDA Draft Guidance for Infusion Pumps [14] now requires pump manufacturers to provide an assurance case with their regulatory submissions.

These activities indicate the significant concerns that FDA has regarding pump safety, and they provide an impetus for research in the areas of software engineering, safety, security, and verification & validation applied to pump development; research which we hope to enable to some extent with the requirements document described here.

III. The Requirements Document

A. Sources of Information

What sources of information were used in the construction of the requirements document? These requirements simulate the result of domain experts working with systems engineers to define function that will be safe for patients, and effective for some medical need. For PCA, that medical need is to provide narcotics to dull excruciating pain. Delivering medication as prescribed is what makes a PCA pump effective. Avoiding overdose, and all other harms to patients, is what makes a PCA pump safe.

These simulated requirements are provided as a public-domain example, because real requirements are highly-confidential to medical device manufacturers, often using proprietary clinical data. However, it should be noted that the authors *are not* clinical experts in PCA infusion therapy. Our primary sources of information were the FDA's guidance documents on infusion pumps (in particular, the description of hazards for pumps) [14], the FDA Infusion Pump Infusion Pump Initiative [13], earlier versions of requirements from the GIP [9], feedback from FDA engineers, and the first author's previous experiences in the medical device industry. The primary contribution to the research community is a collection, in one place, of relevant domain knowledge sufficient for driving *realistic* research investigations of techniques for developing pumps. Even though we made every attempt to be a clinically accurate as possible, we will not claim to have provided the accuracy or detail sufficient for developing a device for which regulatory approval could be obtained. However, one of our goals is to develop examples of risk assessment artifacts and mock regulatory submissions that would provide futher insight into the regulatory submission and approval process.

B. Methodology

What methodology did we use in the process of writing and organizing the requirements? There are a number of potential sources to appeal to for guiding elicitation and capture of requirements. Traditional Software Requirements Specification (SRS) guidelines such as IEEE-830 are general purpose guidelines and fall short of the methodology and insights needed when dealing with safety-critical systems. Our principle source of inspiration has been the US Federal Aviation Administration (FAA) Requirements Engineering Management Handbook (FAA-REMH) written by Rockwell Collins engineers David Lempia and Steven Miller [3]. FAA-REMH focuses directly on recommended practices for requirements engineering for safety-critical embedded systems and provides illustrations using three systems, including a medical system–an Isolette Thermostat for a neonatal incubator. We found it a reasonable resource as it met two key criteria: 1) it is targeted at safety-critical embedded systems, and 2) it was written by experts in the field.

FAA-REMH lists eleven steps that developers should take in order to "progress from an initial, high-level overview of the system... to a detailed description of its behavioral... requirements." The steps are:

1) Develop the System Overview
2) Identify the System Boundary
3) Develop the Operational Concepts
4) Identify the Environmental Assumptions
5) Develop the Functional Architecture
6) Revise the Architecture to Meet Implementation Constraints
7) Identify System Modes
8) Develop the Detailed Behavior and Performance Requirements
9) Define the Software Requirements
10) Allocate System Requirements to Subsystems
11) Provide Rationale

C. Document Structure

What is the overall structure and content of the document? The document has twelve sections, briefly summarized here, with more important sections further elaborated below.

1) **Introduction**: purpose, references, terms, and acronyms
2) **System Overview**: clinical background and need, synopsis, context, external interactions, goals, boundary, and monitored/controlled variables
3) **System Operational Concepts**: use and exception cases
4) **PCA Pump Function**: functional requirements; basal flow rate, patient- or clinician-requested bolus
5) **PCA Pump Interfaces**: interface requirements; sensors, actuators, alarms, control panel, logging, network, reservoir, drug library, scanner
6) **Safety Requirements**: safety architecture, anomaly detection and response, power, diagnostics, tamper-resistance, biocompatibility
7) **Security Requirements**: authentication, confidentiality, provisioning
8) **Requirements Allocation**: requirements must be allocated to functional architecture or labeling
9) **Labeling of Nonfunctional Requirements**: environmental assumptions for use
10) **Functional Architecture**: decomposition of system into functional components
11) **Initialization and Configuration**: simplified set-up instructions
12) **Rationale**: reasons for requirements

D. System Operational Concepts

FAA-REMH recommends creating use cases and exception cases during requirements development to capture system operational concepts. Our requirements document contains roughly 15 pages of use cases. Use cases and exception cases are listed in Table I and Table II, respectively. In addition to a Clinician or Patient, an Integrated Clinical Environment (ICE) compatible Medical Application (called an "App" for short), can also be a system actor [15], [7].

As an example, use case UC1 is presented next. Note that workflow steps, which involve interactions between the Clinician and some agent external to the PCA pump, are marked with WF).

Use Case: Use PCA Pump (UC1)
Related System Goals: G1 and G2

TABLE I
SELECTED USE CASES

ID	Actor	Title	Description
UC1	Clinician	Use PCA Pump	Summary Use Case covering: initialization, attachement, basal infusion, detatchment
UC2	Patient	Administer Patient-Requested Bolus	Extra dose upon patient-determined need.
UC3	Clinician	Administer Clinician-Requested Bolus	Extra dose upon clinician-determined need.
UC4	ICE App	Switch to KVO Infusion Rate	Upon ICE-Detected hazard, switch to KVO infusion rate.
UC5	Clinician	Resume Operation After Hazard	Resume prescribed infusion after clinician determines it is safe.

Primary Actor: Clinician
Secondary Actor: Patient
Precondition:

- Patient is ready for infusion.
- Physician has prescribed drug.
- Pharmacy has filled prescription.
- Pharmacy has installed drug library into PCA pump.
- Drug has been delivered to clinician.
- PCA pump is off.

Main Success Scenario

1) Clinician turns on PCA pump.
2) System sucessfully completes Power-On Self Test and sounds audible alarm. **Exception Case:** *Power-On Self Test Failure*.
3) Clinician acknowledges that the alarm sound is audible. **Exception Case:** *Sound Failure*.
4) Clinician provides identifying information (*e.g.*, scans badge).
5) System confirms that the Clinician is authorized to operate PCA pump for 5 minutes (Δ_{auth}). **Exception Case:** *Clinician Authentication Failure*.
6) Clinician enters patient information.
7) System confirms that patient is authorized to receive medical care for 5 minutes (Δ_{auth}). **Exception Case:** *Patient Authentication Failure*.
8) Clinician provides drug information and patient's prescription from drug container (vial).
9) System confirms that the prescription has originated from an authorized pharmacist. **Exception Case:** *Prescription Authentication Failure*.
10) System ensures that drug information (*e.g.*, limits) is available from its drug library. **Exception Case:** *Drug Library Soft Limit and Drug Library Hard Limit and Drug Library Not Available*.
11) System unlocks reservoir door.
12) Clinician puts drug vial into the reservoir and closes the door.
13) System dectects door closure and presence of drug vial. System locks reservoir door and terminates clinician pump access rights.

WF) Clinician attaches infusion tube and needle to pump.

14) Clinician primes pump. **Exception Case:** *Pump Priming Failure*.
15) System confirms that pump is primed.

WF) Clinician inserts infusion needle into patient's vein.

16) Clinician indicates that basal-rate infusion should be begun.
17) System infuses drug at basal rate.
18) System processes Bolus dose requests: see **Use Cases** UC2 and UC3.
19) Clinician indicates that infusion should be stopped.
20) System stops infusion.

WF) Clinician removes infusion needle from patient's vein.

21) Clinician provides identifying information (*e.g.*, scans badge).
22) System confirms that the Clinician is authorized to operate PCA pump for 5 minutes (Δ_{auth}). **Exception Case:** *Clinician Authentication Failure*.
23) System unlocks reservoir door.

WF) Clinician removes drug vial, closes the door, returning remaining drug to pharmacy.

24) System dectects door closure and absense of drug vial. System locks reservoir door and terminates clinician pump access rights.
25) Clinician turns off PCA pump.

We do not claim to provide an exhaustive collection of use cases. We aimed for a level of coverage that would allow us to establish traceability of each of the subsequent requirements to one or more use cases.

E. Individual Requirements

Each non-functional requirement, or functional requirement that is cross-cutting (relative to the Use Case Model), has its own paragraph, which allows unique numbering and tracing to functional architecture components responsible for implementing them. There are subsections of statements for functions such as *Basal Flow Rate*, *Patient-Requested Bolus*, *Clinician-Requested Bolus*. Requirements are given for the actions of pump sensors (for detecting flow rate, occlusion of drug delivery tubes, and air-in-line embolism (bubbles)), actuators (the pumping action itself), and alarms. Requirements are also given for supporting functionality such as the clinician interface, on-device drug library, logging, and drug reservoir. Safety and security are broken out into their own distinct sections.

Finally, an important goal of our work is the investigation of issues associated with network-enabled interoperable devices that conform to the ICE architecture [7]. Accordingly, the document gives some initial requirements related to the ICE interface of the device (we expect these to evolve as our research progresses).

Here are some samples of requirements from the *patient bolus* function.

- A *patient-requested bolus* shall be delivered at its prescribed rate, F_{bolus}, in addition to the prescribed basal flow rate, F_{basal}, but no more than the maximum flow rate for the pump, F_{max}.
- Patient-requested bolus shall not be delivered more often than a prescribed number of minutes, Δ_{prb}.
- Prescribed $VTBI$ and rate shall not exceed the hard limits set by the drug library from the hospital pharmacy for the drug loaded in the PCA pump.
- Patient-requested bolus shall *not* be delivered if infusing prescribed $VTBI$ will exceed hard limits retrieved from the drug library for the volume of drug infused over a period of time. Pump rate shall be reduced to KVO and a *max dose warning* be issued.

TABLE II
SELECTED PCA EXCEPTION CASES

ID	Actor	Title	Description
EC1	Patient or Clinician	Bolus Request Too Soon	bolus request denied because minimum time between boluses had not elapsed
EC2	Clinician	Drug Library Soft Limit	basal rate or bolus VTBI exceeded soft limit
EC3	Clinician	Drug Library Hard Limit	basal rate or bolus VTBI exceeded hard limit
EC4		Power-On Self Test Failure	power-on self test fails
EC5		Internal Electronic Failure	PCA pump detects its own failure
EC6	Clinician	Pump Priming Failure	pump fails to prime after loading drug reservoir
EC7		Over-Flow Rate Alarm	measured flow rate exceeds setting
EC8		Under-Flow Rate Alarm	measured flow rate below setting
EC9		Pump Overheating	pump temperature exceeds 55 C
EC10		Downstream Occlusion	blockage between pump and patient
EC11		Upstream Occlusion	blockage between reservoir and pump
EC12		Air-in-line Embolism	bubble detection
EC13		Maximum Safe Dose	dose reaches maximum allowed by drug library
EC14	Clinician	Clinician Authentication Failure	clinician not authorized to operate pump
EC15	Clinician	Patient Authentication Failure	patient not admitted to hospital
EC16	Clinician	Prescription Authentication Failure	drug or prescription not intended for this patient
EC17	Clinician	Sound Failure	no audible alarm
EC18		ICE Failure	indication of no ICE alarms enabled
EC19		Drug Library Not Available	the drug library fails authenticity or integrity check

- Patient-requested bolus delivery shall be immediately halted when alarms sound.

Here are some from the audible alarm safety requirements.

- Alarms shall cause *audible alarms signals* that meet the requirements of Tables 203 and 204 of standard IEC 60601-1-8 for alarm pulses, bursts, and harmonics.
- The *auditory volume* of audible alarms signals shall conform to Section 201.3.3.2 *Volume of auditory ALARM SIGNALS and INFORMATION SIGNALS* of standard IEC 60601-1-8.
- The *alarm melody* of audible alarms signals shall conform to Table AAA.1 of standard IEC 60601-1-8 for drug or fluid delivery. "C d g" shall be used for medium priority alarms; "C d g - C d" shall be used for high priority alarms; "e c" shall be used for warnings and low priority alarms.[2]
- Each tone in the alarm melody shall be composed of a minimum of 4 *harmonic components* in the range 300 Hz to 4000 Hz comprising an inverted 9th jazz chord.
- Temporarily paused alarms shall reactivate $\Delta_{ap} = 10$ minutes after inactivation.

F. Functional Architecture

Following the REMH methodology, the PCA Pump *functional architecture* partitions system operation into smaller, simpler pieces, recursively. The PCA function is partitioned into the functional components in Table III, depicted in Figure 2.

G. Safety Architecture

Another distinguishing feature of our requirements work is the illustration of a *safety architecture*. This is a notion that the FDA wishes to expose and illustrate to the academic and industrial communities. A medical device safety architecture is a hardware and software subsystem, separate from that which performs the normal operations of the device. The subsystem *detects* potential safety hazards, *acts* to prevent or mitigate a detected hazard, *notifies* a person that a hazard was detected, and *records* its occurrence for later investigation. In the case of a PCA pump, the safety subsystem detects faults that may harm the patient (e.g., improper flow rate, air bubbles in tube, etc.), signals an alarm or warning, and stops infusion or reduces infusion to a keep vein open rate depending on the fault(s) detected. The components in the safety system are listed in Table IV, and depicted in Figure 3.

TABLE III
FUNCTIONAL COMPONENTS

Component	Behavior
fluid	holds and moves drug
operation	controls pump operation
safety	checks for faults; inhibits possibly hazardous infusion; signals alarms and warnings
power	coordinates battery and power supply; detects power anomalies

TABLE IV
SAFETY COMPONENTS

Component	Behavior
pump_fault_manager	handles pump fault signals
alarm_process	holds thread which controls alarms
fault_logger	record faults
error_detector	handle hardware-detected faults
failure_led	indicates hardware failure

[2]The characters c, d, e, g, C refer to relative musical pitches and C is one octave above c.

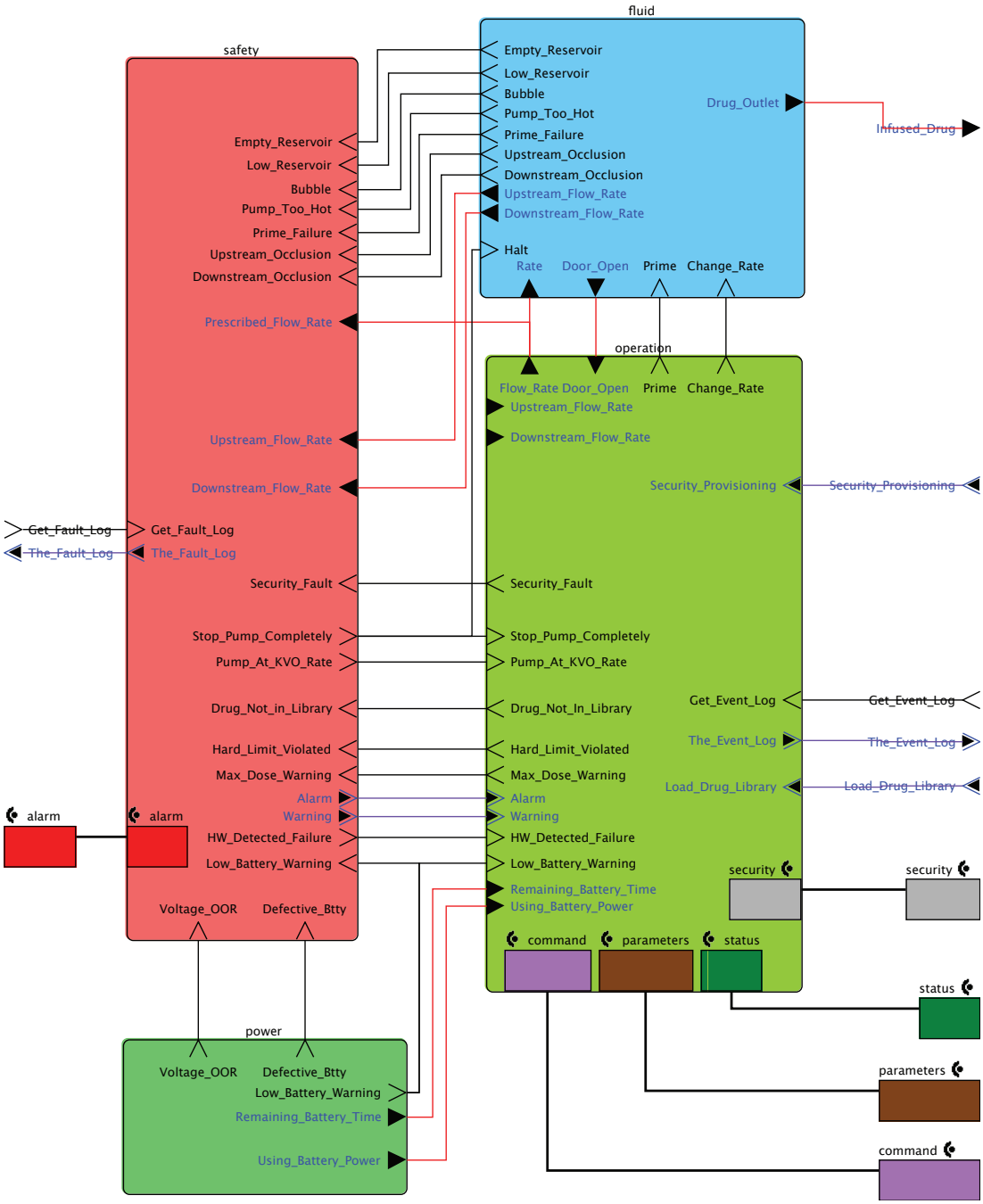

Fig. 2. PCA Functional Components

TABLE V
TOP-LEVEL COMPONENTS

Component	Behavior
ICE Bus Adaptor	translates events and data into bus transactions
PCA	performs pump operation
Maintenance	technician access to logs, drug library

IV. ONGOING AND FUTURE WORK

The PCA requirements document forms the basis for creation of interrelated design artifacts to serve as public examples of model-based engineering of safety-critical medical devices.

The AADL model of the PCA Pump is being augmented with behaviors defined using Behavioral Language for Embedded Systems with Software (BLESS) annex subclauses and specifications using BLESS Assertion properties [16]. The

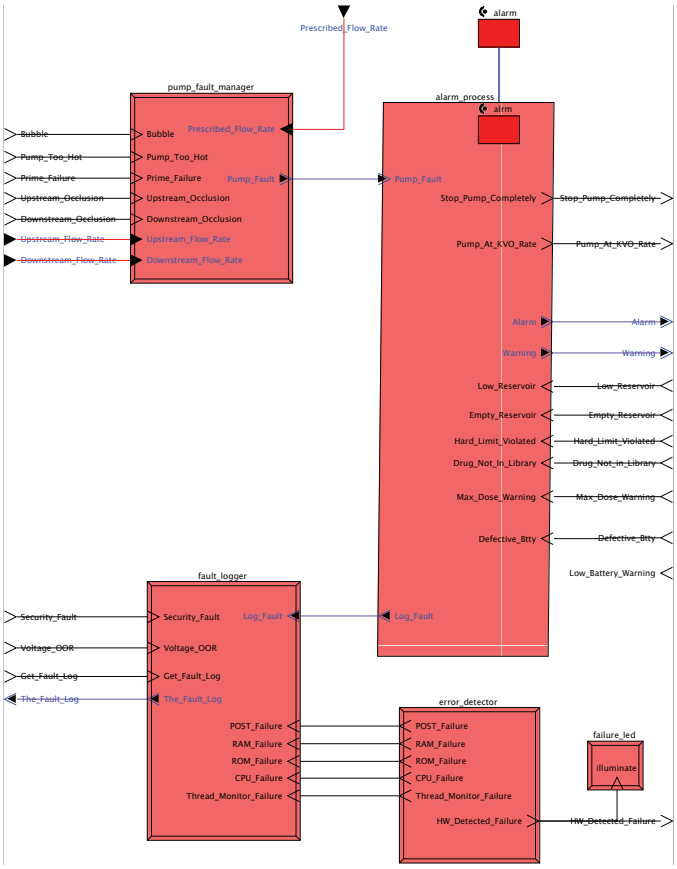

Fig. 3. Safety Subsystem

goal of this work is compositional correctness proofs of high-level safety and efficacy properties.

We plan to add Error Modeling annex subclauses (EMV2) [17] to model fault initiation, error transmission and transformation, and failure occurrence. Failure modes and effect analysis (FMEA), fault-tree analysis (FTA), and other reliability or safety analyses can be applied to the architectural model. We also plan to link requirements to architectural elements with the Requirements Definition and Analysis Language (RDAL) [18] used by the RDAL Tool Environment (RDALTE) plug-in to the Open-Source AADL Tool Environment (OSATE) [5] provided by the Software Engineering Institute of Carnegie Mellon University. The assurance case for the PCA pump will trace to RDAL for evidence, and then to implementing architectural components and verification artifacts like tests and proofs.

The PCA pump model exemplifies safety architecture in [19]. Prototype hardware is being designed by Kansas State's Electrical and Computer Engineering department. The model is subject for security architecture development, and definition of interoperability using AADL's polymorphic type checking of architectural components.

Under the NSF FDA Scholar-In-Residence (SIR) program, we regularly meet with the U.S. FDA Office of Science and Engineering Laboratories (OSEL) to report research progress and receive guidance to ensure FDA can commend the artifacts as examples of good design. Assurance cases, required in submissions for FDA approval of infusion pumps, must be kept confidential to protect manufacturer's trade secrets, thus cannot be used as examples of clear and convincing arguments that a pump is both safe and effective. Hopefully, the public PCA pump design artifacts will help manufacturers write applications for approval that FDA's Office of Device Evaluation can easily and quickly ascertain that argument from evidence convincingly demonstrates the device(s) will be acceptably safe and effective.

Acknowledgements

This work is supported in part by the National Science Foundation under Grants #0932289, 1065887, 1238431, 1239543 and by the NIH/NIBIB Quantum program.

References

[1] "PCA Pump Requirements and Architecture," info.santoslab.org/-research/pca, 2013.
[2] Boston Scientific, "PACEMAKER system requirements specification," http://sqrl.mcmaster.ca/pacemaker.htm, 2007.
[3] D. Lempia and S. Miller, "DOT/FAA/AR-08/32. requirements engineering management handbook," Federal Aviation Administration, 2009.
[4] P. H. Feiler and D. P. Gluch, *Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language*. Addison-Wesley, 2013.
[5] "Architecture Analysis & Design Language," www.aadl.info, 2013.
[6] B. R. Larson, P. Chalin, and J. Hatcliff, "BLESS: Formal specification and verification of behaviors for embedded systems with software," in *NASA Formal Methods Conference*, 2013 (submitted for publication).
[7] *ASTM F2761-2009. Medical Devices and Medical Systems — Essential Safety Requirements for Equipment Comprising the Patient-Centric Integrated Clinical Environment (ICE), Part 1: General Requirements and Conceptual Model*, ASTM International, 2009.
[8] "The Software Certification Consortium website," www.cps-vo.org, 2013.
[9] "Generic Infusion Pump Project Homepage," http://rtg.cis.upenn.edu/gip.php3.
[10] D. Arney, R. Jetley, P. Jones, I. Lee, and O. Sokolsky, "Formal methods based development of a PCA infusion pump reference model: Generic Infusion Pump (GIP) project," in *Proceedings of 2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability*, jun 2007.
[11] R. W. Hicks, V. Sikirica, W. Nelson, J. R. Schein, and D. D. Cousins, "Medication errors involving patient-controlled analgesia," *American Journal of Health-System Pharmacy*, vol. 65, no. 5, pp. 429–440, March 2008.
[12] J. Commission, "Preventing patient-controlled analgesia overdose," *Joint Commission Perspectives on Patient Safety*, p. 11, October 2005.
[13] "US FDA Infusion Pump Improvement Initiative," April 2010.
[14] "Guidance for Industry and FDA Staff - Total Product Life Cycle: Infusion Pump - Premarket Notification [510(k)] Submissions (Draft Guidance)," http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm206153.htm, 2010.
[15] B. Larson, J. Hatcliff, S. Procter, and P. Chalin, "Requirements specification for apps in medical application platforms," in *4th International Workshop on Software Engineering in Health Care (SEHC)*. IEEE, 2012, pp. 26–32.
[16] B. R. Larson, "Behavior language for embedded systems with software annex sublanguage for aadl," info.santoslab.org/research/aadl/bless, 2013.
[17] *AADL Error Model Annex*, SAE International, 2013.
[18] *AADL Requirements Definition and Analysis Language*, SAE International, 2013.
[19] B. R. Larson, P. Jones, and Y. Zhang, "Medical device safety architecture," Kansas State and US FDA, Amherst, MA, USA, Tech. Rep., 2013.