2017 International Conference on Engineering of Complex Computer Systems

Study of Software-Related Causes in the FDA Medical Device Recalls

Zhicheng Fu, Chunhui Guo, Zhenyu Zhang,
Shangping Ren
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
{zfu11, cguo13, zzhang111}@hawk.iit.edu,
ren@iit.edu

Yu Jiang
School of Software
Tsinghua University
Beijing, China
jy1989@mail.tsinghua.edu.cn

Lui Sha
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
lrs@illinois.edu

***Abstract*—As technology advances, medical devices are playing increasingly more important roles in patient care. Unfortunately, based on the U.S. Food and Drug Administration (FDA) data, medical device recalls are at an all time high. One of the major causes of the recalls is due to defective software. In fact, one in every three medical devices that use software for operation has been recalled because of failures in the software itself. Unlike traditional software, software-based medical devices have specific domain fault modes, and these fault modes have been not addressed in software design literature, such as dosage calculation fault. In this paper, we first present a process that collects software-related medical device recalls from the FDA database. Collecting all software-related medical device recalls is an effort that needs the support and contributions from a large research, industrial, and medical community, To facility such effort, we have developed a web-based platform for different users to contribute and share new software-related medical device recalls into the collection. Second, we analyze one hundred software-related recalls that we have collected from the FDA database. Our analysis reveals that there are four major categories of software failures in medical device recalls and implicit assumptions made by medical device manufacturers are among one of the leading causes in medical device recalls. Last, we present an approach for implicit assumption management in medical cyber-physical system designs.**

***Keywords*—medical device recalls analysis, medical cyber-physical systems, software fault modes, root causes, implicit assumptions**

I. Introduction

Medical devices are often subject to a large number of failures with potentially catastrophic impacts on patients. Based on the U.S. Food and Drug Administration (FDA) data, from 2006 to 2011, there were 5,294 recalls and 1,154,451 adverse events resulting in 92,600 patient injuries and 25,800 deaths [1]. The FDA data also reveals that one in every three medical devices that use software for operation has been recalled due to failures in software itself [2]. We use the following medical device recall to explain how software failures have caused a medical device recall.

FDA Medical Device Recall 1. ***Medtronic Recalls SynchroMed II and SynchroMed EL Implantable Drug Infusion Pumps, May 14, 2013 [3].*** *Medtronic is recalling the SynchroMed Implantable Infusion Pumps because a software problem may cause unintended delivery of drugs during a priming bolus procedure, used to quickly deliver large dose of medication from the device to the patients spine. During this procedure, patients may receive the drug unintentionally at a high rate of infusion in the cerebrospinal fluid followed by a period of reduced drug delivery after the priming bolus. This can result in a drug overdose or underdose which can lead to serious adverse health consequences such as respiratory depression, coma or death.*

This recall reveals a fact that unlike traditional software systems, software-based medical devices have some specific domain fault modes which have been not addressed in software design literature, such as dosage calculation errors in the recall. Hence, studying the root causes of software failures in medical device recalls is critical in understanding the problem and enhance the areas in current medical cyber-physical system design and development.

The FDA classifies recalls into three classes based on the relative degree of health hazards a medical device presents, i.e., Class I, Class II and Class III with Class I for the most severe hazards. It has released an analysis about the distributions of these three medical device recall classes [2]. However, the FDA analysis does not reveal the root causes of software failures in the recalls, but being able to identify the root causes of the failures is critical in addressing the failures and preventing future recalls.

In this paper, we present a process to collect software-related medical device recalls from the FDA database. Collecting all software-related medical device recalls is an effort that needs the support and contributions from a large research, industrial, and medical community. To facility such effort, we develop a web-based platform [4] that enables users to contribute and share new software-related medical device recalls. In addition, we classify major categories of software failures which have most frequently occurred in medical domains and conduct an analysis on these recalls to determine the leading causes of these recalls. The analysis reveals that implicit assumptions are one of the root causes of medical device recalls. Hence, being able to explicitly and accurately specify assumptions and integrate these assumptions in medical cyber-physical system (M-CPS) design and development is critical

978-1-5386-2431-9/17 $31.00 © 2017 IEEE
DOI 10.1109/ICECCS.2017.20

to ensure the safety of M-CPS. For that, we introduce an approach that models and integrates assumptions in M-CPS design to improve the safety of M-CPS.

The rest of the paper is organized as following: we introduce the background of the FDA medical device recall database in Section II. Section III describes the procedure of how we collect software-related recalls from the FDA database. Section IV defines the major categories of software failures and illustrates the analysis of software-related recalls. In Section V, we conduct analysis of different types of assumptions in software-related recalls and present an approach to explicitly model and integrate assumptions into M-CPS design. We draw conclusions and point out future work in Section VI.

II. FDA Medical Device Recall Database

The FDA regulates medical devices sold in the U.S. by requiring manufacturers to follow a set of pre- and post-market regulatory controls. The FDA has classified and described over 1,700 distinct types of devices and organized them with the Code of Federal Regulations into 16 medical specialties "panels" such as Cardiovascular devices or Ear, Nose, and Throat devices [5]. After a medical device is distributed in the market, the FDA monitors reports of adverse events and other problems with the device and, when necessary, alerts health professionals and the public to ensure proper use of the device and safety of patients.

The FDAs Recalls database contains medical device recalls since November 1, 2002 [2]. A recall is a voluntary action that a manufacturer, distributor, or other responsible party takes to correct or remove from the market any medical device that violates the laws administrated by the FDA. Recalls are initiated to protect the public health and well-being from devices that are defective or that present health risks such as disease, injury, or death. In rare cases, if the company fails to voluntarily recall a device that presents a health risk, the FDA might issue a recall order to the manufacturer.

The FDA classifies recalls into three classes based on the relative degree of health hazard a device presents. Class I recalls indicate that there is a reasonable chance that use of the device will cause serious adverse health problems or death. Class II indicates devices that might cause temporary or medically reversible adverse health consequences or pose a remote chance of serious health problems. Class III indicates that devices violate the laws administrated by the FDA but are not likely to cause adverse health consequences.

In the FDA database, a medical device recall entry has the following information: 1) recall title, 2) recall class type, 3) recall posted date, 4) recall number, 5) device name, 6) recalling firm, 7) reasons of the recall, 8) action of the recall, 9) instructions for recovery and 10) device distribution. The *reasons of the recall* contains human-written, unstructured text explaining the main causes of the recall.

III. Software-Related Medical Device Recall Collection

To identify the recalls that are possibly due to software issues from the FDA database, we first need to identify whether the reasons for a recall contains semantic-related software keywords, then manually review the reason and communicate with its recalling firms to confirm whether it is a software-related recall. Fig. 1 shows the workflow that how we collect software-related medical device recalls from the FDA medical device recall database.

At the beginning, we extract all the medical device recalls reported to the FDA between 1 January 2014 and 31 December 2016, and store the result as **Recall Record V1**. For each recall in **Recall Record V1**, we use a Part-Of-Speech Tagger (POS Tagger) [6] to marking up each words in the reason field of the recall as nouns, verbs, adjectives, adverbs, etc. For example, the reason of Simens's Picture Archiving and Communication System recall [7] is "RGB images will show?, since the calculation of HU is not possible". After tagging process, the sentence is presented as "RGB**/NN** images**/NN** will**/RB** show**/VB** ?**/.** ,**/,** since**/CC** the**/DT** calculation**/NN** of**/IN** HU**/NN** is**/VB** not**/RB** possible**/JJ**", where the meanings of tags are shown in Table I.

TABLE I
Notation of Tags

Tag	Description	Example
CC	Coordin. Conjunction	since
DT	Determiner	the
IN	Preposition	of
JJ	Adjective	possible
NN	Noun	images
RB	Adverb	not
VB	Verb	show
.	Sentence-final punc	(. ? !)
,	Comma	,

Based on the tagging results, we calculate the number of times a word occurs in the recalls, which is called as *term frequency*. However, in the previous example, the term "the" is so common that term frequency will tend to incorrectly emphasize recalls which happen to use the word "the", and fail to give enough weight to the more meaningful term "calculation". In other words, the term "the" is not a good keyword to distinguish relevant or non-relevant software-related recalls, while the less common word "calculation" does.

The term frequencyinverse document frequency (TF_IDF) [8] is one of the most popular term-weighting schemes intending to reflect how important a word is in a document. Hence, rather than using simple *term frequency* to identify important terms, we apply the TF_IDF method to extract the most relevant nouns and adjectives from the recalls, and reduce the weight of terms (such as "the") that occurs frequently.

To determine which noun or adjective is semantically related to software in the medical domain, we need to define a set of software-related keywords in the medical domain as a comparison objective. For collecting and analyzing software-related medical device recalls, we have developed a web-based

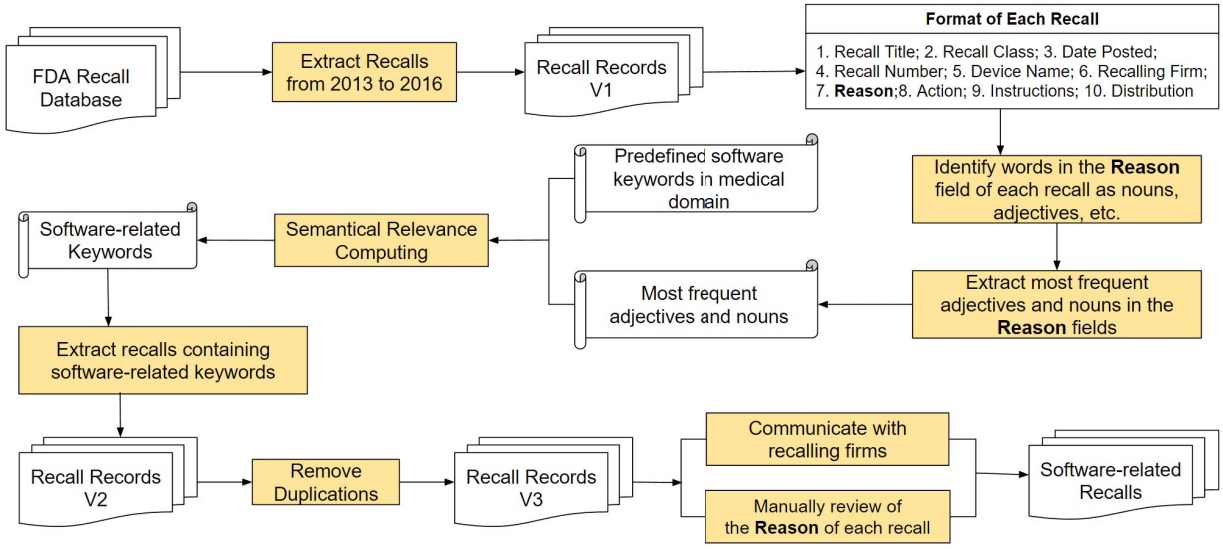

Fig. 1. Flow for Collecting Software-Related Medical Device Recalls

application [4] that allows users to upload software-related medical device recalls, and provide the ability that allows users to use a few keywords to tag the recalls. By extracting the tagging words from our website, we have collected a set of software-keywords, as shown in Table II.

TABLE II
PRE-DEFINED SOFTWARE-RELATED KEYWORDS

Pre-defined Software-related Keywords
system, software, application, database, program, integration, display, function, code, bug, error, fail, verification, validation, self-test, reboot, web, robotic, calculation, document, performance, workstation, expected, senor, alarm, message, screen, signal, interface, monitor, button, key, network, terminal, model, mode, communication, interaction, battery, power, supply, outlet, plug, power-up, discharge, charger, pause, terminate, dosage, environment

We can then calculate the semantic relevance of words from most relevant nouns and adjectives by comparing with the pre-defined software-related keywords. Algorithm 1 computes the semantic relevance and retrieves software-related nouns and adjectives.

Algorithm 1 generates a set of most frequently used software-related words. We use keywords matching to extract the medical device recalls whose reason fields contain software-related words identified by the algorithm. The extracted records are denoted as **Recall Record V2**. As many of the recall records may have the same reasons because the same components or parts are used in different devices or models manufactured by the same company, we use recall title and recalling firm as a basis to remove duplicated entries and get

Algorithm 1 S-RKEYWORDS(W, W_{pre})

Input: A set of words W, and the pre-defined set of software-related keywords W_{pre}.
Output: The set of software-related words W_{out}.

```
Use word2vec model [9] to map each word w_i ∈ W and w_j ∈ W_pre to vector representation as vec(w_i) and vec(w_j)
for each work w_j in W_pre do
    if there exists a word w_i ∈ W, cos(vec(w_j), vec(w_i)) ≥ threshold then
        Put w_i to W_out
    end if
end for
return W_out
```

another set of medical device recalls as **Recall Record V3**.

By manually reviewing each recall in **Recall Record V3**, we exclude the records whose reason for the recalls do not indicate they are software-related recalls. In addition, we also communicate the recalling firms about unclear reasons of the recalls and request more details about the recalls, such as requesting detailed recall letters. At the end, we finalize a list of software-related recalls as **Software-related Recalls**, which can be accessed through http://gauss.cs.iit.edu/~code/recalls.html.

To utilize **Software-related Recalls** and provide a platform for users freely participating in adding new recalls, tagging reasons of the recalls, querying and analyzing the recalls, we have developed a social networking service based web application http://gauss.cs.iit.edu/~code/recalls.html. The application front-end interface is organized around three main interconnected visualization panels: 1) the recalls display, 2) the modal panel for uploading new recalls, and 3) the querying panel. All three panels are interactive and allow users to

navigate intuitively among them.

The recall display panel, shown in Fig. 2 allows the user to see at a glance the latest recalls posted by others. Each recall

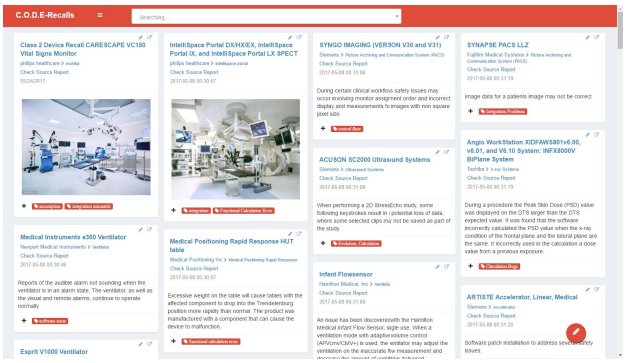

Fig. 2. Home Page Display of the Application

contains: 1) recall title, 2) recalling firm, 3) device type, 4) original URL, 5) date posted and 6) keywords to represent the recall. All these information is required to post a new recall to the application as shown in Fig. 3.

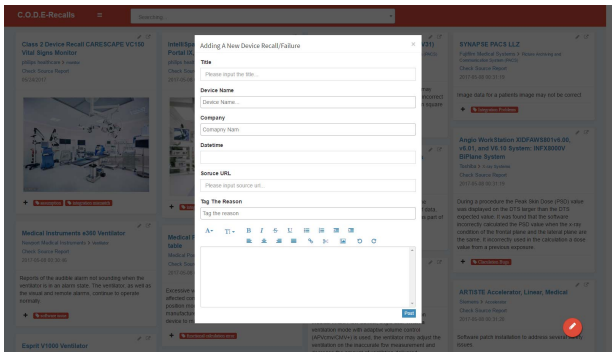

Fig. 3. Uploading New Recall

As shown in Fig. 2, we use red icons to represent the tagging keywords of each recall. The modification of the tagging words can be processed through the red plus icon. In addition, the application allows users to filter recalls by tagging words, as well as providing different querying methods, such as querying by tagging words, device name, reason of recalls or recalling firm name, as shown in Fig. 4. Through this online free platform, users not only can capture, share and analyze the medical device recalls, but also can find interesting motivation examples to drive their research directions.

IV. Software-Related Medical Device Recalls Analysis

We use 100 identified software-related recalls as the basis for deriving statistics on fault categories of software-related failures. Before analyzing the recalls based on software-related fault categories, we illustrate the distribution of the recalls across different risk classes defined by the FDA, shown in

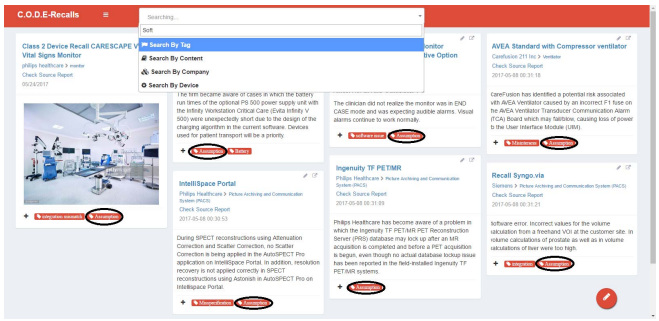

Fig. 4. Query Interfaces and Results

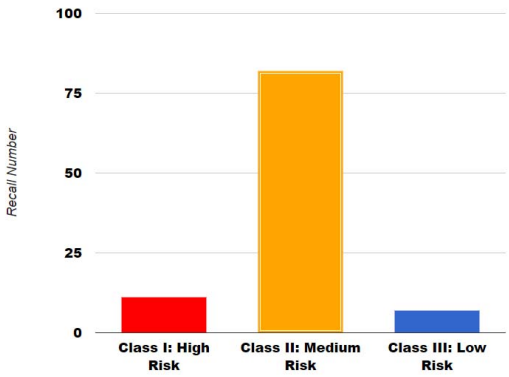

Fig. 5. Distribution of software-related recalls in the FDA's risk levels

Fig. 5. Fig. 5 82% of software-related recalls are classified as class II, with a medium risk of health consequences. However, the FDA's risk level itself can not reflect the root causes of software-related medical device recalls. To give more detailed information about the failures that might impact the safe functioning of a software-based medical device, we further group the failures under four categories:

- Control Flow Fault
- Calculation Fault
- System Integration Fault
- Human-Machine Interaction Fault

Fig. 6 illustrates the distribution of recalls across different fault categories. The majority (93%) of software-related recalls were classified into these four fault categories, while the rest of the recalls, whose descriptions indicate they are software-related failures but do not clearly belong to any of the four categories, are classified as *Other* category.

1) Control Flow Fault: Medical treatment scenarios are often complicated resulting in complicated control flow in medical systems, which increase the probability of resulting in control flow fault. For example, the medical treatment guideline [10] for stroke contains more than 30 main steps, and the complicated execution orders of the steps make control flow more error-prone. We group the failures of control flow into five fields: 1) Inconsistent logics from requirements; 2) Exception handling fault; 3) Block or unblock interrupts fault; 4) Execution orders of functions fault; and 5) Conditional

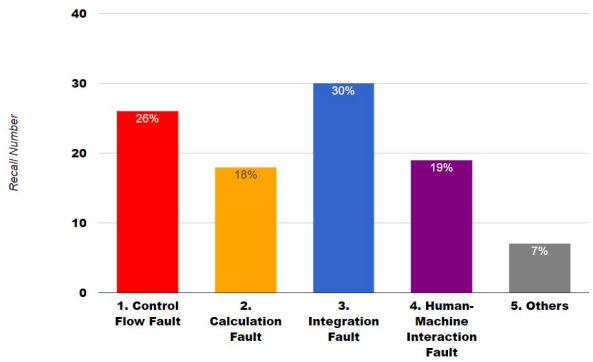

Fig. 6. Distribution of 100 Recalls across Fault Categories

statement fault. Fig. 7 illustrates the distribution of recalls across different fault categories and the followings show these control flow faults, along with a related recall in each field.

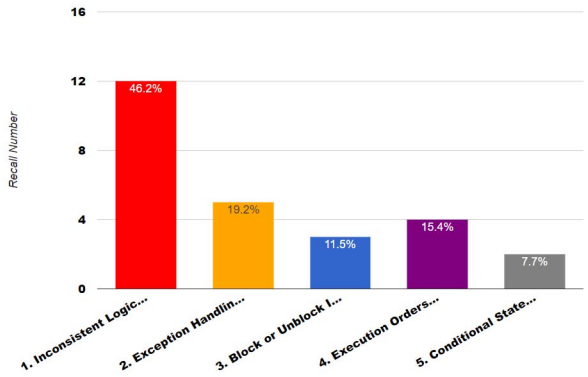

Fig. 7. Distribution of 26 Recalls across Control Flow Fault

- **Inconsistent logic from requirements example: Device Name:** Ventilator, **Date Posted:** 05/09/2016, **Recall Class:** 2, **Recalling Firm:** Hamilton Medical, Inc, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=145040, and **Recall Reason:** After performing the suctioning maneuver, including disconnecting the patient, suctioning , and reconnecting the patient, the preset pattern of ventilation many not continue as expected.
- **Exception handing fault example: Device Name:** Picture Archiving and Communication System, **Date Posted:** 07/08/2014, **Recall Class:** 2, **Recalling Firm:** Philips Healthcare, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=128568, and **Recall Reason:** Faulty Automatic Motion Controller (AMC), a problem in the Power On Self Test (POST) error handling was detected, can result in a hazardous movement of the C-arc. system.
- **Block or unblock interrupts example: Device Name:** Picture Archiving and Communication System, **Date Posted:** 07/08/2014, **Recall Class:** 2, **Recalling Firm:** Siemens, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=133015, and **Recall Reason:** In case of a system crash, images may not be written to the hard disk and this may result in inconsistencies in the database. In case of a system crash (e.g. blue screen, power outage) images may not be written from cache to the hard disk and might get lost.
- **Execution orders of functions fault example: Device Name:** Picture Archiving and Communication System, **Date Posted:** 03/30/2015, **Recall Class:** 2, **Recalling Firm:** Siemens, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=134102, and **Recall Reason:** Possibly incomplete archived studies during pre-fetch. In a server farm setup, when pre-fetch/retrieve operation is performed for partially archived studies, the series that have not yet been archived, will remain unarchived.
- **Conditional statement fault example: Device Name:** Radiation Therapy System, **Date Posted:** 06/21/2014, **Recall Class:** 2, **Recalling Firm:** VARIAN MEDICAL SYSTEMS PARTICLE THERAPY GMBH, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=127498, and **Recall Reason:**Anomaly with the ProBeam System where under certain conditions, the Treatment Control and Monitoring application could fail to send treatment history records to the ARIA database.

2) Calculation Fault: A medical treatment scenario often involves many medicines, which followed by different precisions of dosages, different units of medicines, different data types of dosages. These facts increase the probability of causing calculation faults. We group the failures of calculation into four fields: 1) Incorrect arithmetic/formula; 2) Incorrect/outdated constants; 3) Incorrect conversion; and 4) Incorrect approximation/precision. Fig. 8 illustrates the distribution of recalls across different calculation fields and the followings indicate these calculation faults, along with a related recall in each field.

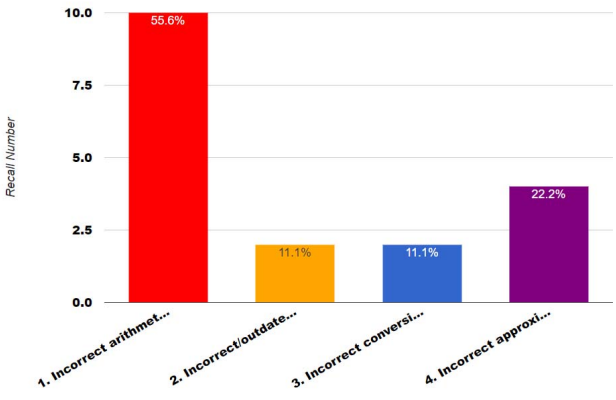

Fig. 8. Distribution of 18 Recalls across Calculation Fault

- **Incorrect arithmetic/formula example: Device Name:** Ventilator, **Date Posted:** 05/09/2014, **Recall**

Class: 2, **Recalling Firm:** Spacelabs Healthcare, **Recall URL:**http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=127487, and **Recall Reason:** Spacelabs Healthcare is voluntarily recalling the Hamilton Galileo Ventilator Flexport, Model 90436A-07, where the monitored Minute Volumes (Vmin) has been reported at one time to reach ten times the actual value on the bedside monitor.

- **Incorrect/outdated constants example: Device Name:** Neurological Stereotaxic Instrument, **Date Posted:** 12/23/2015, **Recall Class:** 2, **Recalling Firm:** Synaptive Medical, Inc, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=142094, and **Recall Reason:** Out of tolerance for radio frequency emissions. At the 150-1000MHz frequency, the testing indicated the BrightMatter Navigation system was up to 20dB uV/meter higher than the applicable IEC 60601-1-2:2007 (Ed3.0) standard specification.
- **Incorrect conversion example: Device Name:** Ultrasound Systems, **Date Posted:** 09/17/2015, **Recall Class:** 2, **Recalling Firm:** Siemens, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=127515, and **Recall Reason:** The ACUSON SC2000 ultrasound system considers uppercase/lowercase differences in the same patient name as unique patient instances when registered on the same ultrasound system. If these differences are not corrected at the time of registration, the system does not capture images or clips.
- **Incorrect approximation/precision example: Device Name:** Picture Archiving and Communication System, **Date Posted:** 02/23/2016, **Recall Class:** 2, **Recalling Firm:** Siemens, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=143694, and **Recall Reason:** Siemens is releasing an updated software version to address several software issues including RGB images will show "?" since calculation of HU is not possible; save as option enabled; changes in access for loading studies; breast region is now properly fitted to segment boundary when clicking fit breast to screen.

3) Integration Fault: A medical treatment scenario often involves many different medical devices, which can increase the complexity of system integration as well as increasing the probability to cause integration fault. We group the failures of integration into four fields: 1) Mismatch of reused component; 2) Mismatch of component interfaces; 3) Inconsistent system evolution; and 4) Mismatch of components configurations. Fig. 9 illustrates the distribution of recalls across different integration fields and the followings represent these integration faults, along with a related recall in each field.

- **Mismatch of reused components example: Device Name:** Picture Archiving and Communication System, **Date Posted:** 03/17/2016, **Recall Class:** 2, **Recalling Firm:** Siemens, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=144270, and **Recall Reason:** Siemens' conducting a recall due to a potential issue when using the measurement package of the VA10 version of syngo Dynamics.

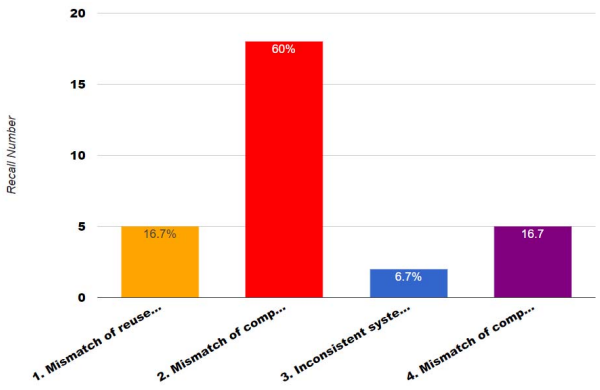

Fig. 9. Distribution of 30 Recalls across Integration Fault

- **Mismatch of components interfaces example: Device Name:** Picture Archiving and Communication System, **Date Posted:** 07/02/2015, **Recall Class:** 2, **Recalling Firm:** Synaptive Medical, Inc, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=138013, and **Recall Reason:** When the E-NMT-01 module is used in conjunction with the ElectroSensor, the Neuromuscular Transmission (NMT) values may indicate a deeper level of muscle relaxation than the actual level of muscle relaxation. In the clinical situation visual movements of the hand are seen after TOF (Train of Four) stimulation, but the patient monitor shows no counts, or counts are not corresponding to the actual value.
- **Inconsistent system evolution example: Device Name:** Intranasal Splint, **Date Posted:** 08/07/2014, **Recall Class:** 2, **Recalling Firm:** Enhancement Medical, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=128288, and **Recall Reason:** Manufacturer made a change in the production process that resulted in a change in final gel weight. RECALL EXPANDED 7/8/2014 Firm expanded their recall to include all lots of product.
- **Mismatch of components configurations example: Device Name:** Monitor, **Date Posted:** 08/05/2014, **Recall Class:** 2, **Recalling Firm:** Curbell Medical, Inc, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=129118, and **Recall Reason:** The firm became aware of a potential problem that was initiated by a customer complaint. After consultation with the manufacturer, it was discovered that a resistor was incorrectly placed within the circuit board on the monitor. This change to the resistor was a planned change to address a product improvement (improve battery drain).

4) Human-Machine Interaction Fault: For medical systems, human-machine interactive behaviors are often performed, such as interactions with a ventilator. Some unexpected interaction pattens cause many integration faults. We

group the failures of human-machine interaction into three fields: 1) Missing/wrong functions of human-machine interactions; 2) Missing/misleading/confusing/error information; and 3) Inappropriate use of keyboard/button. Fig. 10 illustrates the distribution of recalls across different integration fields and the followings show these interaction faults, along with a related recall in each field.

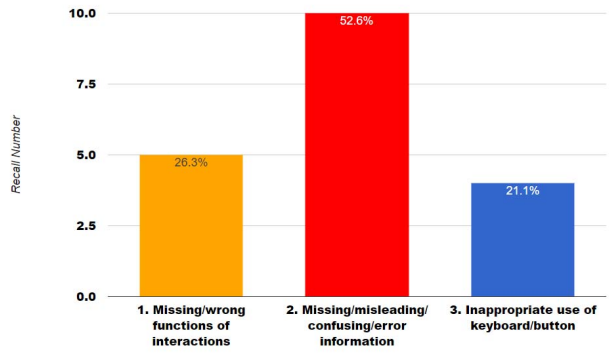

Fig. 10. Distribution of 19 Recalls across Human-Machine Integration Fault

- **Missing/wrong functions of human-machine interactions example: Device Name:** MAMMOMAT Inspiration, **Date Posted:** 04/25/2014, **Recall Class:** 2, **Recalling Firm:** Siemens, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=127169, and **Recall Reason:** There is a potential and possible hazard to the user when using the MAMMOMAT Inspiration PC monitor at the control desk, in that the holder of the PC monitor can break causing an unstable monitor to fall causing possible serious injury.
- **Missing/misleading/confusing/error information example: Device Name:** Ventilator, **Date Posted:** 03/17/2016, **Recall Class:** 2, **Recalling Firm:** Covidien, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=143047, and **Recall Reason:** In the case of a loss of GUI display due to a Backlight Inverter PCBA failure, the ventilator continues to provide uninterrupted ventilatory support at the programmed settings for the patient. However, there is a loss of display and thus there is a necessity to move the patient to another ventilator.
- **Inappropriate use of keyboard/button example: Device Name:** Ventilator, **Date Posted:** 07/16/2015, **Recall Class:** 1, **Recalling Firm:** Breas Medical, **Recall URL:** http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=131484, and **Recall Reason:** Unintended treatment termination could result from a keypad malfunction in some situations. The device erroneously interprets this as a Stop Treatment Instruction. An alarm will not sound, or be registered. Accessories and monitoring equipment connected to the Vivo 50 will stop functioning as the device enters a stand-by mode.

In addition, to see different types of medical devices accounted for the different percentage of device recall events, we category medical device types associated with the recalls shown in Fig. 11. The proportions of medical device types in the recalls is helpful for us to address industry-wide product performance issues and challenges that may impact device quality, safety, and effectiveness.

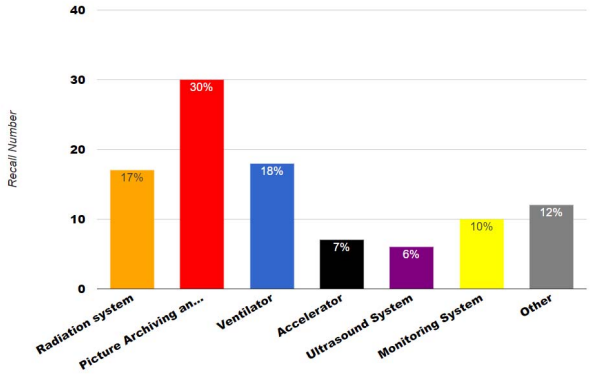

Fig. 11. Distribution of 100 Recalls across Device types

Through manually review all the software-related recalls and their fault categories, we observe some major challenges related to the reliability, safety, and security of medical devices, which has not been addressed in existing works [11], [12], [13]. The following are our insights on some of the future challenges in M-CPS design:

- Develop model-driven design procedures that consider implicit assumptions, flawed requirements, complex software errors, accidents due to dysfunctional interactions among components.
- Integrating patient/doctor/nurse modeling, medical equipments modeling and simulation into M-CPS design.
- Propose safe and efficient strategies that applying advanced techniques such as model checking, comprehensive validation of the system, and run-time monitoring in M-CPS design.

Among the challenges, implicit assumptions are a main factor that is determined to be the cause of failures in safety-critical cyber-physical systems. We use one of the recalls as an example to illustrate how implicit assumptions have caused M-CPS failures.

FDA Medical Device Recall 2. ***Dräger Medical, Evita V500 and Babylog VN500 Ventilators - Faulty Batteries, July 13, 2015 [14].*** *FDA has identified this as a Class I recall, the most serious type of recall. The battery capacity of optional PS500 Power Supply Unit of the Infinity ACS Workstation Critical Care (Evita Infinity V500) did not last as long as expected. The batteries installed in the PS500 depleted much earlier than expected although the battery indicator showed a sufficiently charged battery. Even when the battery depleted totally, the power fail alarm was not generated. If the ventilator shuts down without alarm, a patient may not receive necessary oxygen. This could cause patient injury or death.*

In the recalled ventilator, there are three major components in the system: Controller, Alarm and Battery [15]. The Controller calculates remaining time that the Battery can supply. If the remaining time is less than 30 minutes, the Controller sends an event to the Alarm component to trigger an alarm for medical staffs. Through the initial analysis, we classify the recall reason as a incorrect battery capacity calculation formula. However, with deep analysis, the actual root cause of this failure is that ventilators are assumed to installed in temperature controlled areas, such as hospital rooms, where the temperature is maintained at normal room temperature, i.e. [15C, 35C]. In this environment, the battery can provide full capacity. However if the temperature is not in the assumed range, the battery can only provide partial capacity. This unanticipated change of battery capacity will cause Controller to miscalculate the remaining time and hence fail to send an alarm event on time, which can potentially cause patient injury or death.

In addition, the battery capacity can also be reduced due to unanticipated occurrence of sulfation in the battery. Such unanticipated occurrence of sulfation is caused by users interaction with the ventilator, such as frequent turning on and off the ventilator within a short period of time.

Frequently restarting the ventilator could increase the value of sulfation resulting in unexpected battery capacity reduce. Hence, the charge indicator may not reflect the correct battery capacity. In fact, this failure reveals another implicit assumption is that after the ventilator is powered off, users should wait for at least 3 minutes to startup it again.

Users behavior that frequent powering on and off the ventilator in short time period violates the implicit human-machine interaction assumption, and results in that the battery capacity is unexpectedly reduced.

The system controller is unprepared to handle of this unexpected change and hence fail to send an alarm event before the ventilator is out of power.

The recall shows an inarguable fact that implicit assumptions are one of root causes of software failures in medical device. Hence, how to manage assumptions in M-CPS design is critical to ensure the safety of M-CPS systems.

V. Implicit Assumptions Management in M-CPS Design

In this section, we focus on how to explicitly and accurately specify assumptions and integrate assumptions into M-CPS design to ensure the safety of M-CPS systems. Based on analysis of our recall cases, we introduce the category of assumptions in M-CPS domain. In addition, we provide strategies to model assumptions and integrate assumption models into system models. So that the integrated system models can be validated by both medical and engineering professionals and system safety properties can be formally verified with existing model verification tools.

A. Assumption Categories in M-CPS Design

To better understand how assumptions will contribute to the four fault categories of software failures and classify characters of assumptions in M-CPS design, we categorize assumptions into tree types: 1) physical environment assumptions, 2) human-machine interaction assumptions, and 3) cyber-layer assumptions. Among the 100 software-related medical device recalls, there are 46 recalls related to implicit assumptions. In the assumption-related recalls, 8 recalls, 10 recalls, and 28 recalls are related to physical environment assumptions, human-machine interaction assumptions, and cyber-layer assumptions, respectively, as shown in Fig. 12.

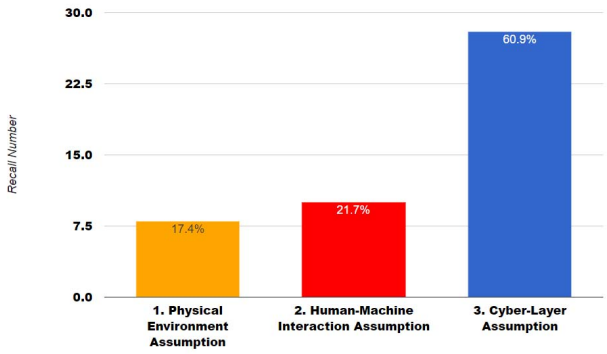

Fig. 12. Distribution of 46 Recalls across Assumption Types

1) Physical Environment Assumptions: In M-CPS, system behaviors are often restricted by the values of system parameters, while the values of system parameters can be affected by physical environment conditions, such as the discharge rate of ventilator battery can be improved with the temperature decrease. We summarize relevant characteristics of physical environment as followings:

1) **Temperature**
   The temperature level might be too high or too low, making batteries in medical devices can not last as expected.
2) **Humidity**
   The humidity level might be too high, making ventilators hard to remove moisture from the air, which could cause patients in danger.
3) **Light**
   The lighting level might be low or high, making it hard to see device displays or controls.
4) **Noise**
   The noise level might be high, making it hard to hear device operation feedback or audible alerts and alarms or to distinguish one alarm from another.
5) **Vibration**
   The device might be used in a moving vehicle, subjecting the device to jostling and vibration that could make it difficult for the user to read a display or perform fine motor movements.
6) **Altitude**
   The altitude level might be too high, making ventilators hard to ventilate expected volume of oxygen as expected, causing patients in danger.

These characteristics should be taken into account during specifying physical environment assumptions, so that M-CPS could be more accommodating of the conditions of physical environment that could affect their use safety.

2) Human-Machine Interaction Assumptions: To have a better identification of the human-machine interaction assumptions in M-CPS design, we empirically provide the following four types of human-machine interaction assumptions. These types are identified through investigating and analyzing more than 100 medical scenarios which containing various human-machine interaction assumptions, such as 10 mechanical ventilation cases [16].

1) **Timing Constraints**
   For a M-CPS, the system can regulate human to perform correct interaction through timing constraints about human-machine interaction assumptions. Timing constraints allow systems to specify the timing conditions for proper human-machine interaction actions to assure the safety of systems.
   Example: Under the cardiac scenario, CPR progress in each round should be less than 2 minutes.
2) **System Parameter Constraints**
   In a M-CPS, human-machine interactive behaviors are often restricted by the values of system parameters, such as parameters representing patients or systems vital information: heart rate, oxygen level, blood flow, respiratory rate and battery capacity. Making the constraints of system parameters explicit in human-machine interaction assumptions is critical to regulating human-machine interactive behaviors with systems.
   Example: After users press the button to shut down a ventilator, if the patient's hemoglobin level is less than 8gm/dL, this shutting down interaction should be ignored by the ventilator.
3) **Execution Order Constraints**
   A M-CPS that interacts with humans is expected to execute human tasks followed by sequence as specified in human-machine interaction assumptions.
   Example: For a medical ventilator, users should set the mode of mechanical ventilation before setting the value of tidal volume.
4) **Synchronization Constraints**
   In a M-CPS, to achieve some system state multiple human-machine interactions from different users are required to be performed at same time. To perform such tasks correctly, the synchronization constraints in the assumption of multiple human-machine interactions should be specified explicitly.
   Example: For a stroke patient in ICU room, in order to get vital information of the patient from Monitoring System, multiple physicians need to perform different interactions with various devices such as infusion pump and ventilator. Synchronization constraints in human-machine interaction assumptions of the ventilator and infusion pump are required to be explicitly specified to navigate users' interactions.

3) Cyber-Layer Assumptions: Developers make many and varied assumptions as they are involving the interpretation of requirements, availability of resources, and types of data. We have characterized cyber assumptions into several types:

1) **System Configuration Assumptions**
   These are assumptions capturing what is expected of the configuration in which the application will operate. Such as, applications are often developed assuming a particular database product and version for data storage, and the configuration assumptions of hardware. For example, the maximum jitter is a characteristic of a particular sensor hardware. Different hardware will guarantee different values of the maximum jitter. The jitter value has to be configured per hardware.
2) **Control Assumptions [17]**
   These are assumptions capturing expected control flow. For example, developers often potentially define the order of methods' execution without formally documented.
3) **Data Assumptions [17]**
   Data assumptions capture what is expected of input or output data. If developers assume data will always have certain characteristics, they can optimize their algorithms and/or data handling code, resulting in improved performance. It is necessary to record these assumptions and verify whether they are valid and consistent with other modules.
4) **Component Interface Assumptions [18]**
   These assumptions capture how a component is expected to be used. For example, the developer assumes that the application will execute from the command line with two required parameters.

B. Modeling and Integrating Assumptions in M-CPS Design

Since the notorious incidents in the 80's involving the Therac 25 radiation therapy machines [19], which was the result of implicit assumptions made in system design, many work has focused on how to efficiently capture and validate cyber-layer assumptions at the interface level of components [20], [17], [18] in auto industry and aerospace. For example, an assumption management framework has been introduced by Tirumala with the aim of designing a set of well-defined vocabularies to encode architectural assumptions of a system [18]. However, few researchers have made efforts on capturing, validating and verifying physical environment and human-machine interaction assumptions in medical cyber-physical system design.

To address the problem, we present an approach that enables engineers focusing on how to explicitly and accurately specify assumptions and integrate the assumptions into M-CPS design to ensure the safety of software-based medical devices. Fig. 13 depicts the high level view of our M-CPS design architecture with assumptions modeling and integrating.

In the medical domain, engineers are more familiar with mathematical structures and operations whereas medical professionals are more used to statecharts as disease and treatment models often have high resemblance to statecharts. To enable

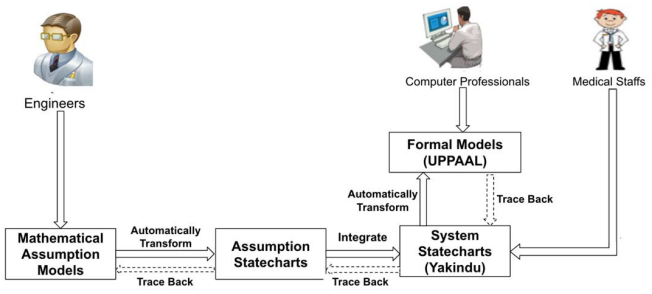

Fig. 13. M-CPS Design Architecture with Assumptions

engineers and medical professionals to communicate about assumptions without misunderstandings and explicitly represent assumptions, our strategy is to define mathematical models for engineers to explicitly and accurately specify assumptions. The mathematical assumption models are then automatically transformed into statechart models and integrated with system statechart models so that the integrated models can be validated by both medical and engineering professionals. The validated models are automatically transformed into formal models to verify safety properties by computer scientists [21]. Applying the approach, we have successfully modeled and integrated physical environment and human-machine interaction assumptions in medical cyber-physical system design [22], [23].

VI. Conclusion and Future Work

In this paper, we present a procedure to collect software-related medical device recalls from the FDA database and develop a web-based platform that enables users to add new and share about software-related medical device recalls. In addition, we classify major categories of software failures most frequently occurred in medical domain and conduct an analysis on these recalls to determine the leading causes of these recalls. The analysis reveals that implicit assumptions is one of the root causes of medical device recall. We introduce an approach explicitly model and integrate physical environment and human-machine interaction assumptions in M-CPS design.

In the future work, we need to collect and study more software-related medical device recalls to find out or confirm the detailed root causes of software failures in medical device recalls. By conducting analysis of the root causes, we hope to identify the potential hazards, safety requirements and assumptions, and risk mitigation techniques and strategies to design the next generation of devices and prevent re-occurrence of similar adverse events in the future.

Acknowledgment

The research is supported in part by NSF CNS 1545008 and NSF CNS 1545002.

References

[1] Homa Alemzadeh, Ravishankar K. Iyer, Zbigniew Kalbarczyk, and Jai Raman. Analysis of safety-critical computer failures in medical devices. *IEEE Security and Privacy*, July 2013.

[2] Food, Drug Administration Center for Devices, Radiological Health, Office of Compliance Division of Analysis, and Program Operations. Medical device recall reportvfy2003 to fy2012. https://www.fda.gov/downloads/aboutfda/centersoffices/officeofmedicalproductsandtobacco/cdrh/cdrhtransparency/ucm388442.pdf, 2013.

[3] U.S. Food and Drug Administration. Medtronic recalls synchromed ii and synchromed el implantable drug infusion pumps due to failure of priming bolus. https://www.fda.gov/MedicalDevices/Safety/ListofRecalls/ucm546558.htm, 2013.

[4] Zhicheng Fu, Chunhui Guo, Shangping Ren, Yu Jiang, and Lui Sha. C.o.d.e recalls-a communication platform for software-related medical device recall. http://gauss.cs.iit.edu/~code/recalls.html, 2017.

[5] U.S. Food and Drug Administration. Device classification panels. https://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/ClassifyYourDevice/ucm051530.htm, 2017.

[6] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, 2003.

[7] U.S. Food and Drug Administration. Syngo plaza picture archiving and communication system recall. https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRes/res.cfm?ID=142876, 2015.

[8] H. Wu, R. Luk, K. Wong, and K. Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems*, 26, 2008.

[9] Tomas Mikolov at Google. Word2vec. https://deeplearning4j.org/word2vec, 2017.

[10] Audebert HJ, Kukla C, Clarmann von Claranau S, Kuhn J, Vatankhah B, Schenkel J, Ickenstein GW, Haberl RL, and Horn M. Telemedicine for safe and extended use of thrombolysis in stroke: the telemedic pilot project for integrative stroke care (tempis) in bavaria. In *Stroke*, page 287291, 2005.

[11] Yu Jiang, Han Liu, Hui Kong, Rui Wang, Mohammad Hosseini, Jiaguang Sun, and Lui Sha. Use runtime verification to improve the quality of medical care practice. In *2016 38th ACM International Conference on Software Engineering(ICSE)*. ACM, 2016.

[12] Yu Jiang, Houbing Song, Rui Wang, Ming Gu, Jiaguang Sun, and Lui Sha. Data-centered runtime verification of wireless medical cyber-physical system. *IEEE Transactions on Industrial Informatics*, 2016.

[13] Andrew Y-Z Ou, Yu Jiang, Po-Liang Wu, Lui Sha, and Richard B Berlin. Using human intellectual tasks as guidelines to systematically model medical cyber-physical systems. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pages 004394–004399. IEEE, 2016.

[14] U.S. Food and Drug Administration. Medical ventilators - faulty batteries. http://www.fda.gov/MedicalDevices/Safety/ucm460951.htm, 2015.

[15] Cadex Electronics. Battery university. http://batteryuniversity.com/.

[16] University of Washington School of Medicine Mission. Mechanical ventilation cases. https://courses.washington.edu/med610/mechanicalventilation/cases.html, 2016.

[17] Lewis, T. A. G., Mahatham, and Wrage L. Assumptions management in software development. In *Technical Report. CMU*, 2004.

[18] Tirumala A.S. An assumptions management framework for systems software. In *Doctoral Thesis. University of Illinois at Urbana-Champaign*, 2006.

[19] N. Leveson and C. Turner. An investigation of the therac-25 accidents. *IEEE Computer*, 26:1841, July 1993.

[20] Patricia Lago and Hans van Vliet. Explicit assumptions enrich architectural models. In *In Proceedings of the 27th international Conference on Software Engineering*, 2005.

[21] Chunhui Guo, Shangping Ren, Yu Jiang, Po-Liang Wu, Lui Sha, and Richard Berlin. Transforming medical best practice guidelines to executable and verifiable statechart models. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–10, April 2016.

[22] Zhicheng Fu, Chunhui Guo, Shangping Ren, Yu Jiang, and Lui Sha. Modeling and integrating physical environment assumptions in medical cyber-physical system design. In *2017 ACM/IEEE 29th Design, Automation and Test in Europe Conference and Exhibition (DATE)*, April 2017.

[23] Zhicheng Fu, Chunhui Guo, Shangping Ren, Yu Jiang, Yizong Ou, and Lui Sha. Modeling and integrating human interaction assumptions in medical cyber-physical system design. In *The 30th IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, June 2017.