Formal Verification and Safety Assessment of a Hemodialysis Machine*

Shahid Khan[1], Osman Hasan[1], and Atif Mashkoor[2]

[1] School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Islamabad, Pakistan
{shahid.khan1,osman.hasan}@seecs.nust.edu.pk
[2] Software Competence Center Hagenberg GmbH, Hagenberg, Austria
atif.mashkoor@scch.at

Abstract. Given the safety-critical nature of healthcare systems, their rigorous safety assessment, in terms of studying their behavior in the presence of potential faults and how the malfunctioning components cause system failures, is of paramount importance. Traditionally, the safety assessment of a system is done analytically or using simulation based tools. However, the former is prone to human error and the later does not provide a complete analysis, which makes them inappropriate for the safety assessment of healthcare systems. These limitations can be overcome by using formal methods based safety assessment. This paper presents our experience of applying model based safety assessment and system verification tools on a hemodialysis machine. In particular, we use the nuXmv model checker to formally verify a formal model of the given hemodialysis machine. The formal model of the given system is then extended with various fault modes of the system components and the eXtended Safety Assessment Platform is used to check various undesired behaviors of the system using invariant properties defined as Top Level Events. This way, we can automatically generate the FTA and FMEA to do the safety assessment of the given hemodialysis machine.

1 Introduction

Modern healthcare systems are increasingly incorporating computing and communication technologies to provide a safe and reliable experience to the patients in the most effective manner. Given the integration of many technologies and the safety-critical nature of healthcare systems, where a system failure may even result in the loss of human lives, the healthcare system manufacturers and regulatory bodies are obliged to rigorously analyze and control the production and usage of such machines. On the contrary, due to the complex nature of present-age healthcare systems and stringent constraints on their time-to-market, both

* The research presented in this paper is partially supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

healthcare system manufacturers and regulatory bodies have very limited time and resources to perform a thorough safety analysis [22]. For instance, the Food and Drug Administration (FDA) of the USA has to substantively interact with its clients within 90 calender days of the filing date, which is clearly insufficient to perform a detailed analysis of each incoming equipment. The situation is further complicated as the details about the product, submitted for review, typically consist of several hundred pages [17].

Safety assessment of systems mainly involves a set of methods, such as Failure Mode and Effect Analysis (FMEA) [21] and Fault Tree Analysis (FTA) [5], to study the way the faults are dealt-with by the system. FTA is a widely used top down technique, which provides a graphical model for analyzing the conditions and factors causing an undesired Top Level Event (TLE), i.e., a critical event, which can cause the complete system failure upon its occurrence. FMEA, on the other hand, provides a bottom up approach in which atomic low level events are tabulated to check the way they lead to an undesired event.

Traditionally, both FTA and FMEA are done using human interventions. A safety assessment expert along with domain experts enlist the possible failure events and from these events FTA and FMEA are generated and analyzed using paper-and-pencil based analytical techniques. However, the complex nature of the present-age healthcare systems makes their analysis on paper almost impossible. Moreover, such manual analysis is quite prone to human error as well. Alternatively, the failure assessment of complex systems is conducted using simulation tools, such as ReliaSoft[1]. However, the results obtained through these simulation based tools cannot be fully trusted as well due to the involvement of numerical methods and the sampling based nature of simulation, where the given system is not exhaustively tested for all possible scenarios. This inaccuracy limitation makes the simulation based FTA or FMEA infeasible for the safety-critical healthcare systems, where an undetected system fault may lead to the loss of human life in the worst-case scenario.

Formal methods [15], which are computer based mathematical reasoning techniques, have been successfully used to overcome the above-mentioned limitations of the paper-and-pencil proof methods and simulation. The main idea behind the formal analysis of any given system is to first construct a mathematical model of the given system using a state-machine or an appropriate logic and then use logical reasoning and deduction methods to formally verify that this system exhibits the desired characteristics, which are also specified mathematically using an appropriate logic. Formal methods are mainly categorized into two mainstream techniques: 1) Model checking [3] that is a state-based technique in which system behavior, specified as a state-machine, is analyzed by verifying the temporal properties exhaustively over the entire state-space of the formal model of the given system within a computer, and 2) theorem proving [15] that allows using logical reasoning to verify relationships between a system and its properties as theorems, specified in an appropriate logic, using a computer.

Both model checking and theorem proving have been used for the FT-based

[1] ReliaSoft.:`http://www.reliasoft.com/`

failure analysis of many real-world systems such as wheel brake system [10] and satellite solar arrays [1]. To the best of our knowledge, formal safety assessment of healthcare systems has not been reported in the literature so far. We believe that using formal methods for the safety assessment of healthcare systems would not only ensure more accurate results, compared to the traditional simulation and analytical based analysis techniques, but would also allow the manufacturers and regulators to manage the safety assessment of healthcare systems within their resources and time constraints. As a first step towards this direction, we investigate the formal safety assessment of a hemodialysis machine [19], which is used to remove metabolic waste from the blood in case of a kidney failure, making it a very safety-critical machine. The hemodialysis machine is a classical example of cyber-physical system and has been identified as a potential candidate of formal safety analysis of a S# based analysis framework [14]. Another main motivation of choosing a hemodialysis machine as our application is the availability of its detailed description along with the required functional requirements [19] as a case study to promote the usage of formal methods in medical cyber-physical systems. All of the reported work, in response to this case study, focused on the formal specification and/or functional verification of this machine using various formal methods, like Event-B [16, 18], Hybrid Event B [4], Algebraic State Transition Diagrams (ASTD) [12] and Abstract State Machines (ASMs) [2]. Thus, in this paper, we extend these recently reported efforts by presenting the formal safety assessment of this hemodialysis machine.
In particular, we chose to build upon the classical ASM based analysis of the hemodialysis machine [2], in which the ASM model of the hemodialysis machine was automatically translated to the corresponding Symbolic Model Verifier (SMV) model for its functional verification by the nuXmv model checker. In this work, we enhance their SMV model with various failure modes for the safety assessment of the given system using the eXtended Safety Assessment Platform (xSAP) tool [9]. The main motivation behind choosing xSAP and the nuXmv model checker for the proposed safety assessment is the ability to conduct a comprehensive analysis using both FTA and FMEA methods since, to the best of our knowledge, the theorem proving based safety analysis does not support FMEA as of now. Moreover, a distinguishing feature of our work is that a formally verified model of the hemodialysis machine is used to integrate the failure modes and analyze the safety aspects.

2 Preliminaries

2.1 Model Checking and nuXmv Model Checker

Model checking [3] is primarily used as a verification technique for reactive systems, i.e., the systems whose behavior is dependent on time and their environment. The inputs to a model checker include a finite-state model of the system that needs to be analyzed along with the intended system properties, which are expressed in *temporal* logic, which is a logic that allows expressing time-dependent behaviors. The model checker automatically and exhaustively

verifies if the properties hold for the given system while providing an error trace in case of a failing property. The state-space of a system grows exponentially with the increase in the size of system variables and their possible values. Thus, it becomes computationally impossible to explore the entire state-space with limited resources of time and memory for larger models. This problem, termed as *state-space explosion* [3], is usually resolved by using efficient algorithms and techniques, like symbolic [7] and Bounded Model Checking (BMC) [8]. The main idea behind BMC is to allow the model checker to check the given property for a partial model, based on the user provided depth. The model checker detects the failing property if it fails in this reduced model. Otherwise, the depth of BMC is incrementally increased in search of a failing property.
The nuXmv model checker supports a wide range of systems, including the infinite state systems, by introducing the new data types of *integers* and *reals* and using Satisfiability Modulo Theory (SMT) [6] for verification. The system to be verified is modeled in a modular manner using the SMV language [7], which allows declaring of Variables (VAR), macros (DEFINE), environment variables interacting with system (IVAR), state transition relations (using INIT and NEXT statements) and nondeterminism. The properties [11] to be verified can be specified in nuXmv using the Linear Temporal Logic (LTL) or the Computation Tree Logic (CTL). LTL specifications are written in nuXmv with the help of logical operations like, AND (&), OR (|), Exclusive OR (xor), Exclusive NOR (xnor), Implication ($\rightarrow$) Equality ($\leftrightarrow$), and temporal operators, like Globally (G), Finally (F), neXt (X) and Until (U). Similarly, the CTL specifications can be written by combining logical operations with quantified temporal operators, like Exists Globally (EG), Exists neXt state (EX) and for All Finally (AF). In case a property turns out to be false, a counterexample in the execution trace of the state machine is provided. Although the approaches used by nuXmv are in general incomplete, a Lasso-shaped counter example is always found if it is guaranteed to exist [11].

2.2 eXtended Safety Assessment Platform

xSAP [9] is the safety assessment tool supported by the nuXmv [11] model checker. xSAP requires three inputs, i.e., a nominal model written in the SMV language, Fault Extension Instructions (FEI) written in a dedicated FEI language and fault library to perform the safety assessment of a system. The nominal model is written in the SMV language and consists of a modular architecture of the system under investigation along with some additional variables, called affected symbols. The FEI file provides the fault definitions in a SMV understandable format. xSAP uses its built-in fault library, which is also customizable, to interpret the FEI [9]. The FEI file mainly consists of fault slices, where each fault slice targets an affected symbol of a nominal component, which is a module in nominal model. Upon execution, xSAP forces affected symbols to be stuck at some value to emulate the behavior of fault occurrence in the system. Each fault slice represents a single or a set of basic failure modes targeting single affected symbol. Upon construction of the overall system state space, these failure

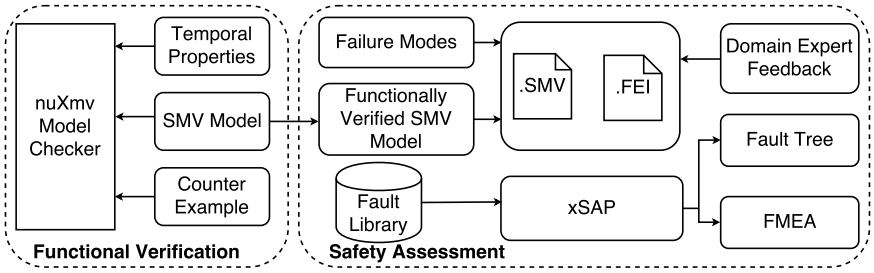

Fig. 1. Proposed Formal Safety Assessment Approach

modes lead to more complex system failures through the mechanism of local and global dynamic models. For the safety assessment of the overall system, TLE are defined as *invariant* properties, which mainly describe the bad behavior of the system. For instance, in the context of the hemodialysis machine, if the system is in the *self test* phase then it is desirable that it eventually successfully completes the self test and goes to the next phase, i.e., *connect concentrate*. The TLE in this case would be `!CN.self_test_status`. Upon execution, xSAP will identify all fault slices and all basic failure modes that can lead to this undesirable behavior, and these fault slices and failure modes are then used to automatically generate the fault tree for the specific event [9]. The xSAP supports many classical tools for safety analysis, including FTA, FMEA, failure propagation analysis using Timed Failure Propagation Graphs (TFPGs), and Common Cause Analysis (CCA). One of the main strengths of this approach is that it automatically generates these artifacts from a formal model, which has been independently checked for its functional correctness using nuXmv.

3 Proposed Approach

The proposed formal analysis approach for healthcare systems, depicted in Fig. 1, is divided in two phases, i.e., Formal Functional Verification (FFV) phase and Formal Safety Assessment (FSA) phase. The phase of functional verification requires a SMV model of the given healthcare system and the associated *temporal* properties capturing the functional requirements. The nuXmv model checker exhaustively checks the model against the provided *temporal* properties and provides the counterexamples in case of failing properties. These counterexamples can then be investigated to check whether the problem is due to a modeling error or actually a functional bug in the system. The modeling issues can be rectified by iteratively refining the SMV model to remove all issues until all the properties are successfully verified. On the other hand, the system designers can be consulted in case of identifying a design bug. Thus, upon the completion of the FFV phase, we obtain a functionally verified SMV model against all its requirements. We use this model in the FSA phase to introduce the affected symbols and provide the fault extension in the .FEI file. Besides the above-mentioned inputs, we consider the involvement of domain experts in this step very important

as they can provide useful insights in the modeling process and greatly facilitate the fault identification due to their past experiences in the domain. Both of these files, i.e., the .SMV file containing the nominal model and the .FEI file containing fault extension instructions, are provided to xSAP for model extension as mentioned in Section 2.2. The xSAP extends the provided .SMV model based on the information provided in the .FEI by invoking its fault library, and applies the fault slices written in the .FEI file on the .SMV nominal component affected symbols of the .SMV file. The next step is to provide the TLE along with this extended model to xSAP to perform the safety assessment. The xSAP automatically generates the FTA and the FMEA tables satisfying TLEs. These artifacts can be subsequently documented and further analyzed for the safety assessment of the given healthcare system.

4 Hemodialysis Machine

Hemodialysis machines are used to remove a controlled amount of metabolic wastes from blood in the case of kidney failures. Their correct operation is the key for the patients wellbeing and thus they can be classified as a safety-critical healthcare system. The machine's internal architecture, as depicted in Fig. 2, can be mainly divided into 8 sub-blocks. Each block further consists of various components having predefined functions. A brief description of each block and its constituent components is given below

Low Level and High Level Controller The controller module [19] consist of two sub-modules, i.e., high level and low level controllers. The former mainly interacts between the machine and the operator through a Graphical User Interface (GUI). Moreover, it also connects the machine with the cyberspace to facilitate remote therapy and on-line observation of therapy results. Whereas, the low level controller acts as a coordinator of tasks between the remaining modules of the machine and thus plays an important role for the successful operation of the overall machine. It receives feedback from different sensors and transmits actuation signals to fulfill the requirements of the machine. For our proposed safety assessment, we have considered the low level controller only. This is because ensuring the cybersecurity is in itself a major challenge and considering it here would divert the focus of this paper to general cybersecurity issues rather than the safety assessment of healthcare systems.

Extracorporeal Blood Circuit (EBC) This module connects the patient to the machine through the Arterial and Vascular (AV) connections. It consists of 2 Venous Peristaltic Pumps (VPP), 1 Arterial Peristaltic Pump (APP), 1 Blood Pump (BP), 1 Heparin Syringe Pump (HSP), a Disposable System (DS) (connectors, drip chambers, tubing), a Safety Air Detector (SAD), 2 Pressure Transducers (1 for Venous (VPT) and other for Arterial (APT) side) and 1 Venous Valve (VV).

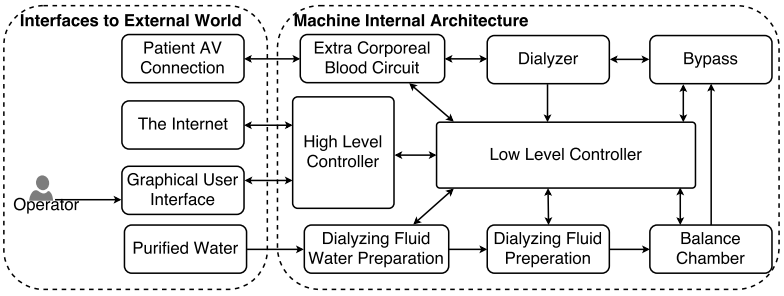

Fig. 2. Hemodialysis Machine Architecture

Dialyzer This module mainly performs the dialysis of patient's blood. It consists of a bidirectional diffusive membrane, which filters out a predetermined amount of the metabolic wastes from the blood.

Bypass It bypasses the dialyzing process when the temperature raises beyond a certain limit or an out-of-proportion concentration of acid and/or bicarbonate is detected in the Dialyzing Fluid (DF). The bypass module mainly consists of two Valves (V1 and V2).

Balance Chamber The Balance Chamber (BC) keeps a balance between the incoming and outgoing DF. It consist of two chambers with a flexible membrane and two Magnetic Position Sensors (MPS1 and MPS2) to keep track of the flexible membrane position.

Dialyzer Fluid Preparation This module is mainly responsible for mixing the prepared water with acid and bicarbonate concentrates. It consists of a Conductivity Meter (CM) and a Temperature Sensor (TC) to monitor the parameters of the prepared water.

Dialyzer Fluid Water Preparation (DFWP) The DFWP mainly degasses and heats the refined water and subsequently provides the processed water to the DF preparation module. It comprises of a Degassing Chamber (DC), a Heater (HT) and a Reverse Osmosis (RO) filter.

Failure modes Now, we describe the failures of the hemodialysis machine [10] [13] [2]. These failures are mainly associated with the modules and sub-modules described above. As explained in Section 3, these failure modes are first expressed in the FEI file and then integrated with the control logic of the machine to emulate run-time feedback and controlling actuation signal mechanism.

[2] Courtesy: Fresenius Medical Care.: `url:http://fmcna.com//`)

The faults occurring in the *pump* module include permanently being in the off state, not reaching the maximum speed at the maximum voltage, the pump is turning in the wrong direction, the signal of the optical tachometer is going out of range, analog voltage going out of range, rotor turning when it is not supposed to, and pump rate and its setting not being synchronized. The behavior of all pumps, including peristaltic pumps, the heparin pump and blood pump, is captured through the module named *pump*, which is instantiated six times, namely *EBC.APP1*, *EBC.APP2*, *EBC.VPP*, *EBC.HSP* and *BC.UFP* to represent 2 arterial peristaltic pumps, 1 venous peristaltic pump, 1 heparin syringe pump of EBC and 1 ultra filtration pump of the balance chamber, respectively.
The faults occurring in the *disposable tubing system* module include a leak, kinking, clotting and clamping, the fibre clotting of dialyzer, and a closed line. This module is instantiated twice, namely *EBC.DS* and *D.DS*, to represent the disposable system of *EBC* and the disposable assembly of the dialyzer, respectively.
The failure modes of the *valve* module include a failure when the valve is open, failure when it is close, failure at the last commanded position and a failure at an erroneous position. These failure modules are instantiated three times, namely *EBC.VV*, *B.V1*, *B.V2*, to represent the venous valve of the *EBC* and the valve 1 and 2 of the *bypass* module, respectively.
The failure modes of the *chamber* module captures the failing behavior of all chambers, including the balance and degassing chamber, by considering the conditions of low and high fluid levels and low and high pressures. The *chamber* module is instantiated three times, namely *EBC.VC*, *EBC.AC* and *DWP.DC*, to represent the *venous chamber* of the *EBC*, *arterial chamber* of *EBC* and *degassing chamber* of *DWP* stage, respectively.
The conductivity *Meter* module of the DF preparation stage is modeled by undetected erroneous data and no data faults. It is used once, i,e., *DFP.CM*, to represent the *conductivity meter* of DFP module.
The failure modes of sensors associated with temperature, safety air detector and magnetic position are captured by the undetected erroneous data, no data, signal ramping down and signal out of limit events. The *sensor* module is instantiated four times, namely *EBC.SAD*, *BC.MPS1*, *BC.MPS2*, *DFP.TS*, to represent the *air detector* of *EBC* and the magnetic position Sensors 1 and 2 of the balance chamber and the temperature sensor of the DF preparation modules, respectively.
The *Heater* module is required for water heating in the DF water preparation stage. Its failures are captured by the insulation break, burn out of the heating element and malfunctioning or complete failure of heater events. The heater module is instantiated once to represent the heater of dialyzing water preparation stage (*DWP.HT*).
The failure mode of the *Transducer* module is represented by the wet transducer protector (protectors are used to keep interior of pressure transducers from getting wet), obstructed monitor line, erroneous data, no data and data out of limit events. This module is instantiated twice, i.e., *EBC.APT* and *EBC.VPT*, to represent the arterial and the pressure transducers of the EBC module, respectively.

5 Formal Functional Verification and Safety Assessment

The model, described in Section 4, is used to conduct the formal functional verification and safety assessment of the hemodialysis machine using the approach outlined in Section 3. For verification purposes, we used Version 1.0.0 of nuXmv with an *Intel(R) Core(TM) i5-3320M CPU @ 2.60GHZ, x64-based processor*. While the safety assessment is carried out using Version 1.1.0 of xSAP. All reported properties are verified using BMC with a depth of 100. However, the TLEs are exhaustively checked for developing of the fault trees and FMEA. Next, we describe four top level events for which we generated the fault trees and FMEA during the safety assessment process[3].

5.1 Self Test Pass

We first verify that there is at least one instance when the self test of the hemodialysis machine, i.e., `CN.prepPhase = SELF_TEST`, during the Preparation Phase succeeds and the system goes to the next Preparation Phase, i.e., Connecting Concentrate [19], `CN.prepPhase = CONNECT_CONCENTRATE`. The CTL property used to check this property is as follows
`AG(CN.prepPhase = SELF_TEST -> EF CN.prepPhase = CONNECT_CONCENTRATE)`. It is important to note that the property is not verified for all cases since it obviously fails in the presence of machine faults. To generate the FT and FMEA for the bad conditions, i.e., when the system is stuck at self test, we introduced an undefined state, i.e., `PREP_UNDEF`. Such that, the system goes in this state whenever it is not in any known state of the Preparation Phase. The TLE is `CN.prepPhase = PREP_UNDEF`. The verification of this property allows us to automatically generate the fault tree, which is partially shown in Fig. 3a. It can be clearly seen that a fault occurrence in any component can lead to an overall system failure. The reliability of the overall system can be increased by introducing redundancy in the system components. Thus, we added another venous peristaltic pump *EBC.VPP2* in the system. This change would lead to the addition of an AND gate between both arterial peristaltic pumps, *EBC.VPP1 and EBC.VPP2 (newly introduced in system)*. Which means that, both arterial peristaltic pump 1 and arterial peristaltic pump 2 have to fail simultaneously to lead to the system level failure.
The effect of adding redundancy on the system reliability is further illustrated in Figs. 4b and 4a. In these figures, the horizontal axis represents the failure probability of individual failure events, while the vertical axis depicts the failure probability of subcomponents and the overall system. As discussed in Section 4, the SMV model of hemodialysis machine consists of multiple instances of 8 basic components. The failure probability of an individual event is assigned to every instance of the respective component and the collective failure behavior is computed using the corresponding fault tree. For example, there are five instances of the *pump* module, in the machine, namely, *EBC.APP1*, *EBC.VPP*,

[3] The codes and associated properties are available at: `http://save.seecs.nust.edu.pk/projects/fvsahm/`

EBC.HSP, *BC.UFP* and *BC.BP*. Each instance can fail independently of the other and their collective failing behavior is presented in the graph. Likewise, there are 1, 3, 4, 1, 4, 1, 2 instances of *Disposable_system*, *Valve*, *Chamber*, *Meter*, *Sensor*, *Heater* and *Transducer*, respectively. The failure probability of the individual components is swept from 0 to 1 and the complete behavior of the failure of the hemodialysis system and its constituent components is captured in both figures. It is evident from the figures that the failure probability of the system with redundant components reaches 1 when failure probability of constituent basic events is below 0.1. Whereas, in the case of redundant components, the same probability is around 0.8 when the basic events failure probabilities are between 0.3 and 0.4. The decrease in slope with redundancy implies that for any given basic event failure probability, the likelihood of failure of a system with redundancy is less than the likelihood of failure of a system without redundancy. The relationship between cut-sets of FT and FMEA, as generated by xSAP, is elaborated in Fig. 3e, in the context of self test TLE. The graph is shown on a semilogarithmic scale to suppress the huge difference between FTA and FMEA cut-set values. These statistics were generated from the model having 21 basic failure events. According to the statistics displayed, there are obviously zero cut-sets with cardinality 0. While, the cut-sets with cardinality 1 for both FTA and FMEA are reported by xSAP to be 21 (note that FTA of the system without redundancy had 22 cut-sets with cardinality 1 and no cut-sets for higher cardinalities). This change (from 22 to 21) in cut sets by adding redundancy effects the system reliability as depicted in Fig. 4b). When the cardinality is increased to 2, the number of FT generated cut-sets decreases to 1, but the number of cut-sets reported by FMEA increases to 274. Upon further increasing the cardinality, we reach a stage where no cut-sets for FT were found. Whereas, for FMEA, the number of cut-sets further increases to 2045 and 10900 for cardinalities 3 and 4, respectively. During the formal safety analysis of the hemodialysis machine, it is observed that the number of cut-sets for FMEA reported by xSAP are generally greater than those of FT for the same property and cardinality. This is because the FMEA tables do not present minimal cut-sets leading to TLE, like fault trees. On the contrary, they consider all possible faults even if the faults are not contributing directly to TLE [9].

5.2 Temperature Control

We verify that *if the system is in the preparation phase and performs priming or rinsing or if the system is in the initiation phase, then the dialysate temperature shall remain between 33° C and 40° C* [20].

```
G ((CN.phase = PREPARATION & CN.tubingSystemPhase = PRIMING |
CN.prepPhase = RINSE_DIALYZER) | CN.phase = INITIATION ->
CN.current_temp > 33 & CN.current_temp < 40)
```

We transformed this property to an invariant TLE by asserting that the water temperature must always invariantly remain within 33° and 40°C.

```
CN.current_temp > 40 | CN.current_temp < 33
```

The Fault tree for this TLE is shown in Fig. 3b. As can be see from the figure, the temperature violates this condition limit in the presence of one of the seven events,

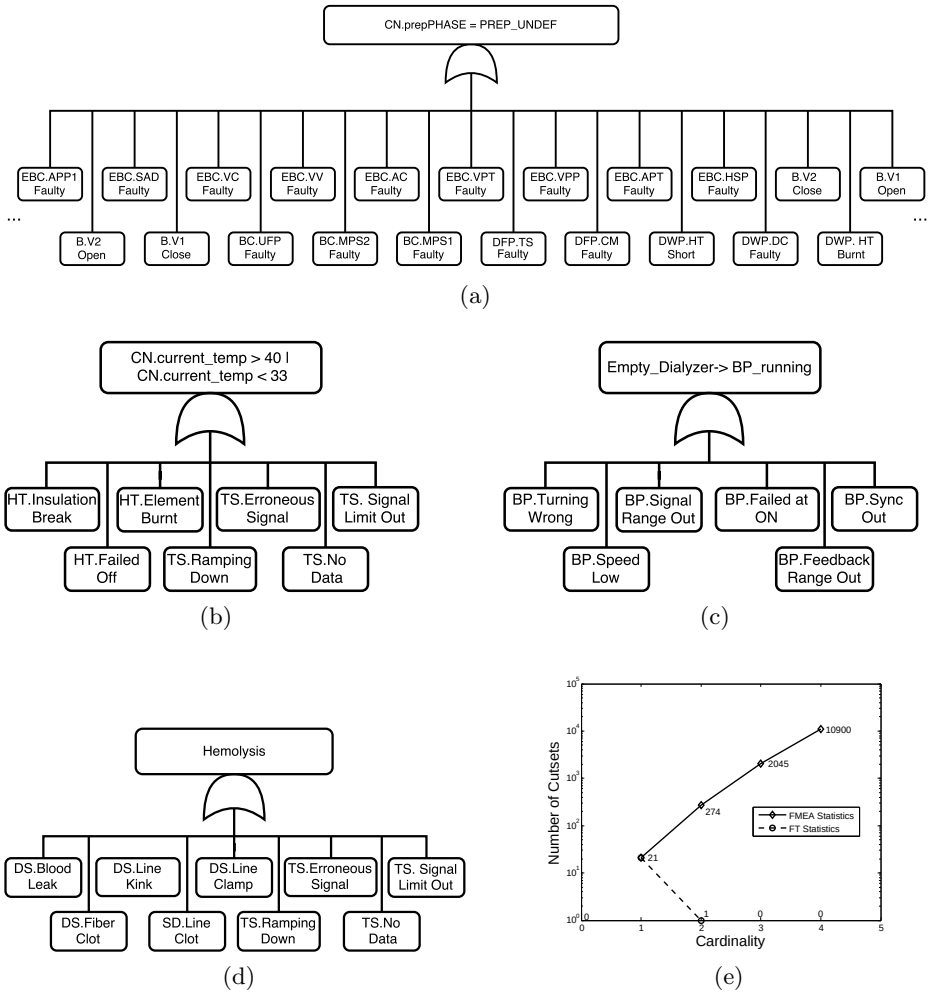

Fig. 3. Fault Trees (a) Self Test of the Hemodialysis Machine (b) Temperature Control (c) Blood Pump Stoppage (d) Hemolysis (e)FMEA and FTA Statistics

namely, the heater insulation gets a short circuit condition or the heater element is blown or the heater fails permanently or the temperature sensor signal ramps down or the sensor signal is out of limit or erroneous or no data can be obtained from the temperature sensor at all.

5.3 Stoppage of Blood Pump

An important safety property for the hemodialysis machine is that its blood pump should stop immediately whenever its dialyzer is empty. The corresponding TLE is
`(TRUE -> CN.dialyzer_empty-> CN.EBC.BP.state = on)`
The fault tree which resulted from the verification of this property is shown in Fig.

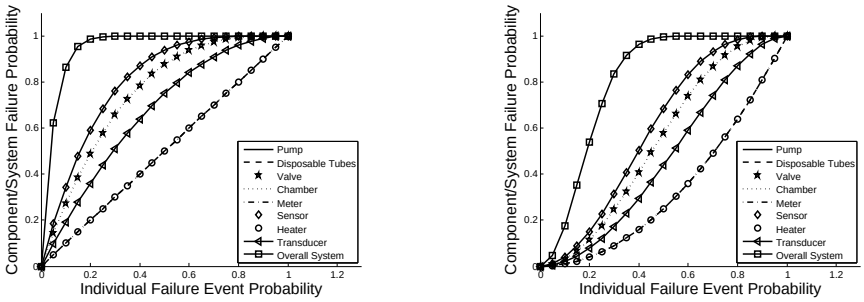

Fig. 4. Hemodialysis System Failure (a) Without Redundancy (b) With Redundancy

3c. Intuitively, this undesired event can occur whenever the blood pump permanently fails at the on condition. Among the failure conditions where pump is not stopping and thus stuck at the on state, as depicted in the failure tree, are the pump turning in the wrong direction, its speed being too low but not halting, the signal of the pump being out of range, the pump explicitly failing at the on condition, the feedback signal of pump being out of range or the pump rotation being out of synchronization.

5.4 Hemolysis

The hemolysis is one of the most undesired conditions that must be averted during hemodialysis. In this condition, the red blood cells are damaged in the dialyzer. The conditions, which can cause the hemolysis are improper flow in the blood lines (due to clamping, kinking, etc.), the dialysate temperature exceeding 42°C, low conductivity of dialysate, high arterial pressure, contaminated dialysate water (contaminations may include bleach, copper or nitrates), and a highly diluted dialysate [13]. As discussed in Section 4, many of these basic events leading to hemolysis are captured in the FEI semantics. We defined a property and named it hemolysis in our code and generated the corresponding fault tree, given in Fig. 3d. The failure events including kinking, clamping, fiber clotting, issues with conductivity meters, heaters, blood pressure are ORed together leading to hemolysis. In case of the emerging remote therapy scenarios, these failure events should be given extreme attention. Moreover, redundancy is strongly recommended here to prevent hemolysis.

6 Conclusion

This paper presents a formal safety assessment approach for a hemodialysis machine. The results obtained from this analysis are quite useful in assessing the safety levels of the hemodialysis machine and thus complement its previously verified functional correctness results. This work can be extended in many directions and one of the possible directions is to further refine the model of the system by adding more architectural details and more detailed failure modes for each device. Another direction is to come up with more interesting safety and security scenarios and check whether the model and thus the system design satisfies those properties or not.

References

1. Ahmad, W., Hasan, O.: Towards Formal Fault Tree Analysis Using Theorem Proving. In: Intelligent Computer Mathematics, LNCS, vol. 9150, pp. 39–54. Springer (2015)
2. Arcaini, P., Bonfanti, S., Gargantini, A., Mashkoor, A., Riccobene, E.: Integrating formal methods into medical software development: The ASM approach. Science of Computer Programming (2017)
3. Baier, C., Katoen, J.P., Larsen, K.G.: Principles of model checking. MIT press (2008)
4. Banach, R.: Hemodialysis Machine in Hybrid Event-B. In: Abstract State Machines, Alloy, B, TLA, VDM, and Z. LNCS, vol. 9675, pp. 376–393. Springer (2016)
5. Barlow, R.E., Chatterjee, P.: Introduction to fault tree analysis. Tech. rep., DTIC Document (1973)
6. Barrett, C.W., Sebastiani, R., Seshia, S.A., Tinelli, C.: Satisfiability Modulo Theories. Handbook of Satisfiability 185, 825–885 (2009)
7. Biere, A., Cimatti, A., Clarke, E.M., Fujita, M., Zhu, Y.: Symbolic Model Checking using SAT Procedures Instead of BDDs. In: Design Automation Conference. pp. 317–320. ACM (1999)
8. Biere, A., Cimatti, A., Clarke, E.M., Strichman, O., Zhu, Y.: Bounded Model Checking. Advances in Computers 58, 117–148 (2003)
9. Bittner, B., Bozzano, M., Cavada, R., Cimatti, A., Gario, M., Griggio, A., Mattarei, C., Micheli, A., Zampedri, G.: The xSAP safety analysis platform. In: Tools and Algorithms for the Construction and Analysis of Systems. LNCS, vol. 9636, pp. 533–539. Springer (2016)
10. Bozzano, M., Cimatti, A., Pires, A.F., Jones, D., Kimberly, G., Petri, T., Robinson, R., Tonetta, S.: Formal design and safety analysis of AIR6110 wheel brake system. In: Computer Aided Verification. LNCS, vol. 9206, pp. 518–535. Springer (2015)
11. Cavada, R., Cimatti, A., Dorigatti, M., Griggio, A., Mariotti, A., Micheli, A., Mover, S., Roveri, M., Tonetta, S.: The nuXmv symbolic model checker. In: Computer Aided Verification. LNCS, vol. 8559, pp. 334–342. Springer (2014)
12. Fayolle, T., Frappier, M., Gervais, F., Laleau, R.: Modelling a Hemodialysis Machine Using Algebraic State-Transition Diagrams and B-like Methods. In: Abstract State Machines, Alloy, B, TLA, VDM, and Z. LNCS, vol. 9675, pp. 394–408. Springer (2016)
13. Fresenius Medical Care: 2008T Hemodialysis Machine, User Manual (2008)
14. Habermaier, A.: Design Time and Run Time Formal Safety Analysis using Executable Models. Ph.D. thesis, University of Augsburg (2016)
15. Hasan, O., Tahar, S.: Formal verification methods. In: Encyclopedia of Information Science and Technology, Third Edition, pp. 7162–7170. IGI Global (2015)
16. Hoang, T.S., Snook, C., Ladenberger, L., Butler, M.: Validating the Requirements and Design of a Hemodialysis Machine Using iUML-B, BMotion Studio, and Co-Simulation. In: Abstract State Machines, Alloy, B, TLA, VDM, and Z. LNCS, vol. 9675, pp. 360–375. Springer (2016)
17. Masci, P., Ayoub, A., Curzon, P., Lee, I., Sokolsky, O., Thimbleby, H.: Model-based development of the generic PCA infusion pump user interface prototype in PVS. In: Computer Safety, Reliability, and Security. LNCS, vol. 8153, pp. 228–240. Springer (2013)
18. Mashkoor, A.: Model-driven development of high-assurance active medical devices. Software Quality Journal 24(3), 571–596 (2016)

19. Mashkoor, A.: The hemodialysis machine case study. In: Abstract State Machines, Alloy, B, TLA, VDM, and Z. LNCS, vol. 9675, pp. 329–343. Springer (2016)
20. Mashkoor, A., Sametinger, J.: Rigorous modeling and analysis of interoperable medical devices. In: Modeling and Simulation in Medicine Symposium. p. 5. Society for Computer Simulation International (2016)
21. Stamatis, D.H.: Failure mode and effect analysis: FMEA from theory to execution. ASQ Quality Press (2003)
22. Zuckerman, D.M., Brown, P., Nissen, S.E.: Medical device recalls and the FDA approval process. Archives of internal medicine 171(11), 1006–1011 (2011)