ISSN: 2634 - 8853

Journal of Engineering and Applied Sciences Technology

SCIENTIFIC Research and Community

Review Article

Open Access

Smart System Modeling and Simulation design of Hemodialysis Machine by SysML with SystemC-AMS

Abbas A Abdulhameed[1], Bassam AlKindy[1] and Basil Al-Mahdawi[2]

[1]Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq

[2]Department of Mechatronics Engineering, Middle Technical University, Baghdad, Iraq

ABSTRACT

The corrections of the large and complex systems have become an issue of great importance recently due to the system development process. The device technology is quite complex and encompasses various disciplines of engineering, medicine, biology, and critical patient-machine interfaces. In this paper, we proposed a simulation approach to the verification and validation of the system of the hemodialysis machine (HD machine) specified by the SysML language. The simulation is carried out through intelligent translation based on the specifications of SysML into SystemC-AMS models. To evaluate and validate the system design, they are translated into a set of formulas and requirements verified when run on the available tools, COSE DA technologies, the WSN simulator developed by SystemC-AMS.

***Corresponding author**

Abbas A Abdulhameed, Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq.

E-mail: abbasabdulazeez@uomustansiriyah.edu.iq

Received: August 18, 2022; **Accepted:** August 30, 2022; **Published:** September 12, 2022

Keywords: SystemC-AMS, SysML, HD, RD, PD, Validation, Modelling, AMD

Introduction

Modeling and engagement in software verification and validation of critical systems are essential to ensure that the implementation of all parts of the software meets required specifications. System verification plays a vital role in achieving modern smart systems in several safety-critical areas such as aviation, modern automated manufacturing, and medical devices accuracy. Nowadays, the use of models is of increasing importance for the design of complex smart systems needed in industrial applications, where these systems are designed to improve their quality in order to ensure their accuracy and high productivity.

The science of systems engineering deals with the entire process of developing complex systems consisting of hardware, software tools, and personnel. These systems are called hybrids, in which some components are characterized by an analogous behavior, while the behavior of others is so discrete [1,2]. As a functioning medical device, according to AMD, any medical equipment that utilizes an electrical energy source or any energy source that is unique from that generated directly by the human body or gravity and can activate by converting this energy is considered AMD [3]. Standalone software can also be considered AMD, according to the Official Journal of the European Union for Medical Devices [4].

The essential goal of the development is that the software takes into account the individual adaptation needs much more reliably than the hardware. A hemodialysis machine (HD machine or dialysis machine) is considered a subclass of AMD for the treatment of kidney failure by filtering the blood and removing toxins from the patient's body [5,6]. During treatment, blood flows through tubes from the body to the dialysis machine. While the blood is in the machine, it passes through a dialyzer filter that removes waste and excess fluid. After the blood has been cleaned, it is returned to the patient's body through other tubes from the dialysis machine. For the experimental validation of our approach in this work in a case study of an HD machine [7].

Systems Engineering (SE) is an interdisciplinary domain of engineering that has appeared as an effective means of designing, integrating, and managing complex systems over the entire system development life cycle (SDLC). SE comprises various software activities that can optimally fulfill the expected purpose. These activities are requirements estimation, verification, functioning analysis, performance analysis, mappings, and specifications of system architecture.

The Model-Based Systems Engineering (MBSE) is a software engineering method specially developed for designing systems that use networked computer models in dealing with basic requirements, the needs of the customer, and the functionality needed in the development define cycle. These requirements include functional and non-functional requirements, system verification, testing, cost, and schedule. In recent decades, MBSE has been developed to become a powerful tool for the analysis and synthesis of complex systems [8]. the System Modeling Language (SysML) is proposed by the Object Management Group

(OMG) [9]. SysML is a universal graphic modeling language for specifying, analyzing, and designing complex systems, hardware, software, information, and processes. In the Unified Modeling Language (UML), SysML is partially adapted to describe the system in the early stages of design with its graphical notation; Nevertheless, SysML is basically a semi-formal language that requires precise semantics to support validation and verification activities.

SystemC is a brand-new powerful approach for machine-degree simulation and modeling language, which gives many benefits to clear up many troubles withinside the machine design [10]. Combining SysML with SystemC lets in a better capability to explain complicated structures out of the SystemC environment's simulation skills through translating the SysML diagrams into SystemC models. Typically, investigating the simulation answers may be a preferred approach of reading the specs for complicated structures.

Simulation is a completely green pedagogy and training method that has been highly advanced these days to be broadly used inside with the training staff for enhancing the abilities of airline pilots and stewards, navy personnel, athletes, and healthcare professionals, such as clinical students. Emulation-primarily based totally training inside the healthcare region can take the shape of screen (computer) emulation, sensible and virtual patients, static and virtual manicures, and activity trainers [11].

Simulation techniques guarantee that a restrained quantity of user-described device paths can also additionally satisfy the required specifications. This paper ambitions to approve the validation of SysML purposeful necessities through translating SysML diagrams into the SystemC surroundings through thinking about the Block Definition Diagram (BDD), Parametric Diagram (PD), and SysML Requirement Diagram (RD). As a result, device fashions cannot be validated as required and in a green way, particularly in the extra complicated structures field.

To cope with this problem, we recommend a transformation approach primarily based totally on Model-Driven Engineering (MDE) to map SysML/PD into the SystemC model [12]. We constitute the SysML necessities through the simulation surroundings SystemC Network Simulator Library (SCNSL) [13]. Then, we examine and confirm the necessities through the use of the CCOSE DA tool. Hence, following this technique might also additionally lead us to locate mistakes at the start of the gadget design.

To decorate the validation of the WSN nonfunctional properties which might be accomplished in general with the aid of using simulation with inside the enterprise domain, we recommend a clever MDE method to layout and validate the WSN power consumption. The verification method, guided with the aid of using the requirement relationships (for instance" satisfy" and" verify" with the version elements), verifies the functional requirements of the state machine diagram. In different words, this modeling approach permits us to lower the design time, decrease the layout cost, and accentuate WSN version efficiency. The constraint block definition diagram and PD are translated toward the SystemC version. The meant translation is modeled with the aid of using a clever transformation and applied in ATL (ATLAS Transformation Language) [14].

The rest of this paper is prepared as follows. In Section 2, we donate the applicable paintings of different researchers in this field. Section 3 offers a historical past at the principles used in this study. In section 4, we state our standard method for verifying the useful necessities of SysML for PD. In addition, it consists of the guideline of thumb translation of PD into SystemC fashions describesribe the extraction simulation (e.g., surroundings to verified houses) from SysML necessities. Sections 5 and 6 reveal the suggested verification method with a case study, which takes into consideration the strategies of the Haemodialysis HD system. Finally, in Section 7, we present the realization and description of ideas for future work.

Related Works

In this work, a compact conceptual model for the engineering of hybrid systems with semantics based on a hybrid extension of EventB is presented. Structural modeling based on the well-known concepts of the entity-relationship model requires only a few extensions of data types and constraints. Conversely, behavioral modeling requires careful separation of synchronous and asynchronous interactions whigh-levelevel means to integrate continuous functions [15]. On this basis, the separation of continuous and hybrid component problems is approached using a sophisticated industrial example of a hemodialysis machine to illustrate the modeling method. A model of the HD Machine Case Study is designed to demonstrate the benefits of the object-oriented functionality of S # for larger models.

The work of suggests how any such scientific tool may be modeled in a manner that automated protection evaluation is feasible and may be included in the improvement life-cycle of a protection-important machine [16]. The hemodialysis device is a bigger case to take a look at with numerous components. First, the machine is decomposed into components, the target is inside the shape of a SysML block definition diagram. Second, the controller of the case modeled the usage supplied of state machines through the S#DSL. The concept of the work is based on the reality that the S# tool chain natively helps the Deductive Cause of Consequence Analysis which is the totally automated version of checking-primarily based on the total protection of the evaluation approach. The latter determines all units of aspect faults with the auto reason a machine risk. Furthermore, S# can supply an approximate estimate of the risk probability. A version is created with a simplified controller of the hemodialysis device and applicable components of its surroundings to carry out a protection evaluation of the usage of Deductive Cause Consequence Analysis [17,18]. The research work of gives a proper specification approach for a hemodialysis device with the use of Event-B [19,20]. The HD device has modeled the use of iUML-B state machines and class diagrams. Next, a corresponding BMotion Studio visualization is constructed and the version is dependent on the usage of refinements to address complexity. The HD device's necessities contain vast sequencing, consumer interactions as properly, and dynamic interaction. Therefore, version validation is done by the use of diagrams to collect the modeling of the sequential properties of the necessities, ProB based animation, and visualization with simulation gear to discover the behavior of the models. Where appropriate, the evidence abilities of Event-B are for use to confirm protection constraints.

Preliminaries

In this section, the SysML modeling language applied in our method for describing the machine after which make use of the formulation in SystemC version simulation environments to designate the necessities of the SystemC software that defines the complicated systems.

SysML Language

SysML is a language of graphical modeling method distinct with an assistant of OMG of semi-formal semantics. Its availability scope improves the UML-primarily based complicated structures improvement approaches with the final success studies from the system engineering discipline. SysML diagrams may divide into 4 pillars: structural, behavior, requirement, and parametric.

The structure pillar gives the hierarchical perspective of the model and offers suggestions concerning the block application, parts, connectors, and ports. The behavior pillar incorporates data flow, relationships, flow activity, and state machine. In addition, this pillar is related to sequence modeling too. The pillars of SysML make the model more steady and complete. Indeed, it gives the chance to study the model through the system from the design viewpoint.

Requirement Diagram

Requirements Diagrams (RD) in the SysML language depict requirements, packages, various classifiers, study cases, rationales, and interconnections. Possible connections that can be handy for those system requirements diagrams are containments, deriveReqt, and requirement dependencies ('Copy', 'Refine', 'Satisfy', 'Trace', and 'Verify'). The clarification notation also can be beneficial in revealing the relationships of different models.

RD could be very beneficial in systematizing the machine's requirements below design with the aid of using explicitly displaying the numerous units of relationships of those explicit requirements. Moreover, the RD has the benefit of standardizing the approach of specifying the requirements via determined semantics. The SysML findings provide support for the model's requirements for machine components being shown, demonstrating that the requirements are part of the machine architecture [21].

RD also provides modeling structures to represent text-based requirements intod integrates them to various modeling components. These requirement modeling components should act as a link between standard requirement management software and SysML models.

Block Definition Diagram

The Block Definition Diagram (BDD) is used to describe the properties and operations of blocks in the case of structural and behavioral attributes. The BDD provides the state and behavior of the system to be represented and is the primary structural element that displays the hierarchies and connections of the system to be depicted. With parts and flow ports making up a block, the physical components are referred to like parts, and the interfaces are known as flow ports [22].

The constraint block is a defining component that defines a Boolean constraint expression. Typically, the expression constraint identified in a constraint block is an equation or an inequality. Mathematical relationship inference can be very useful for enforcing the property value of blocks. These blocks are also useful in the following cases:

1. To confirm validation of system values in an operating system.
2. Performing a technical analysis through a system design phase during the system life cycle.
3. When creating conditions, all variables are called constraint parameters that explain quantities often displayed according to their value types.

Parametric Diagram

The main purpose of the Parametric Diagram (PD) is to highlight the mathematical relationships among system parameters. When using PD, ports stand for restriction parameters, and connections are for bind relationships. Within a constraint block, mathematical relationships are defined as constraint parameters. The constraint properties described by the constraint block take place in the state following other constraint parameters or properties of the blocks. The creation of Block definitions in the model is required for parametric diagrams. Equations are used as constraints on the attributes of these Blocks in parametric definitions. The equations have parameters that are linked to the system's attributes. Constraint Blocks are used in parametric diagrams to define these restrictions. The Block Definition or Internal Block model can be used to derive these. [15].

System C

From various software design languages presented by different studies, SystemC represent an open-source language for designing software program at the design stage. Although this language is primarily based totally on C++, however, it has its own kernel. This layout language has a run-time scheduler that could take care of each of the synchronization and the scheduling of concurrent processes designers the usage of this language may also follow object-orientated competencies to their designed device hardware. SystemC additionally lets in his customers to paintings at a better stage of abstraction for permitting extraordinarily quicker and extra effective architectural tradeoff evaluation of their layout. There are 5 essential software program extensions furnished via way of means of SystemC to version the device hardware [23]:

1. A time notion.
2. Support of Hardware datatypes.
3. Hierarchy modeling and organization.
4. Concurrency.
5. Various communication reliable among processes and modules.

SystemC Simulation Environment

The synchronization and the scheduling of the SystemC processes are taken over by the SystemC kernel simulation. In general, the discrete-event approach employs three data structures: the state variables, the event list, and the clock. The simulation performance depends on two modes of operation: programming routines and event handling.

The programming process is an important task. Over-time, the events are created and classified during the system design process. The importance of the intelligent simulation process in this work is to model the expected specification entity of the system in the form of optimal behavior. This behavior of the considered system entity is encapsulated by the simulation process, which in turn defines the actions that are carried out by this process during the life of the system. The programmer's realization is divided into two influential phases, the "sc start ()" and "sc stop ()".

Usually, "sc start ()" utilized to start the simulation and "sc stop ()" is for termination if the former has no further events to process. A simulation should start with an initialization phase (once the execution of each process) and next fluctuate in the middle of the evaluation and updating phases. The system design evaluation is updated during the evaluation phase in which various processes are carried out and during the update phase in which the values of the signals and other primitive channels are influenced to change the processes. Of the three existing loops, each of which is the result of immediate or timed delta notifications, the scheduling algorithm accounts for these loops as follows:

1. The immediate notification cycle is limited to the evaluation phase.
2. The delta notification cycle takes the path of the evaluation phase, followed by the update phase, and back to the evaluation phase. This cycle continues the simulation in a delta cycle.
3. The route of the assessment phase, followed by the update phase, and back to the assessment phase. This loop increases the simulation time. The order in which the activated processes are executed is undefined, with immediate notification only resulting from a call to ”sc event::notify()” with no argument. A delta notification results from and only from a call to ”sc event::notify()” without delay. A timed notification results only from a call to ”sc event::notify()” with a delay greater than zero.

SystemC Analog/Mixed-Signal (AMS)

SystemC AMS extensions developed after hard work by researchers to extend the capabilities of the modeling language. Using these extensions allows the designer to run simulations and verify software and hardware digital systems and analog/continuous timing.

Therefore, these experiments should pick up on the developments of various industry systems such as in telecommunications, automotive, and semiconductor [29]. These required changes are implemented in a standardized C++ library that follows SystemC's block-oriented methodology. This approach may be used to create transdisciplinary simulation models in the Discrete-Event (DE), Discrete-Time (DT), and Continuous-Time (CT) domains. Accellera Systems Initiative is trying to standardize them so that an efficient framework for constructing embedded AMS systems may be created. A comprehensive specification of the AMS class library can improve the performance of a SystemC AMS solution [27,28].

Standard Architecture for the SystemC AMS Language

Because of the inconsistency associated with complex embedded systems designed today, the responsible designer must combine different techniques and models of computation (MoC) within the system. Thus, the framework of the SystemC AMS standard language, illustrated in Figure 1, was built from multi-layered approach [29].

The Proposed Methodology

The main goal of this study is to present a new method for modeling, simulation, and validation of complex systems. The modeling enables the analysis and validation of the design of new systems. Models can describe the requirements and the structure of the designed system to show the partial or whole properties (i.e. functionality or performance) from the system.

To do so, we define a set of activities corresponding to the proposed validation method by simulating formal SysML specifications. The method is first creates various SysML diagrams (BDD, IBD and PD) using papyrus tools, these diagrams for instance specify the requirements, structure, and the behavior of the complex system, as shown in [24]. Second, based on the MDE, we generate the transformation technique rule using the ATL and Acceleo tools to map the meta-model's SysML diagrams with the meta-model's SystemC models to translate them into the SystemC environment. Finally, we run the SystemC code to run the simulation with the trace to validate the system design. Figure 1 shows a summary of the essential steps of the proposed method.

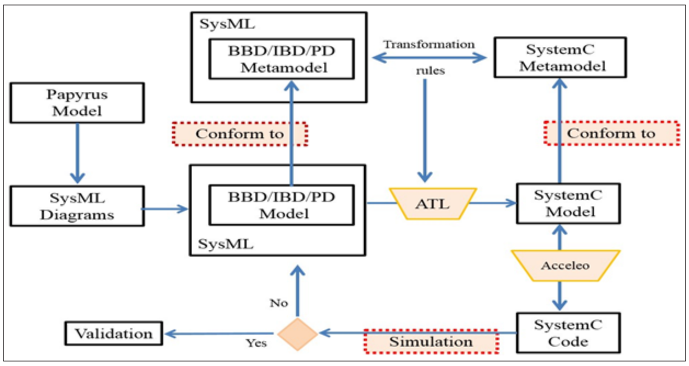

Figure 1: Proposed verification flow from SysML /PD to SystemC-AMS

Meta-Model of SysML

The SysML type is defined on the basis of the core UML on which SysML is constructed, and the SysML metamodel is produced with UML. UML is a concern in the SysML specification. The class diagrams will be utilized to build the SysML metamodel diagram throughout the study. These graphs are identical to those produced by SysML.

The metamodel itself is related to the modeling factors in the SysML about how they're built and the way they're associated with another one. Further, the complete UML metamodel, on which SysML is primarily based totally, is particularly complicated and maybe absolutely impenetrable. The metamodels of the deeply simplified variations of the real metamodel are used to assist the device understanding and to group distinctive stages of the model according to every diagram.

SystemC AMS Meta-Model

The core modules from the "MoC" class are used to structure a SystemC-AMS metamodel. Ports and exports link one module to another. Those two are connected by a channel. An external channel attached to the port must provide an interface for the port. An expert, on the other hand, offers an interface that is processed by the internal channel and connects with the export. As demonstrated in Figure 2, the behavior of a module is determined by one or more simulation processes. All of the system's modular building components are inherited from the class object "Adapter."

Transformation Mapping Rules

By combining SysML with SystemC-AMS, the Papyrus graphical tool was used to build the SysML diagrams. The system description is supplied as the starting point for adding RD, constraint BDD, and PD to the models. The system's requirements and structural details are included in this model specification. The target language, SystemC-AMS, can have a variety of semantic translations with the SystemC-AMS environment's behavioral elements [25]. Some of these translation rules are determined and analyzed. These rules contain the whole hierarchical design by describing the structural perspective through a top-level block SysML constraint BDD and PD. Each composite block is also built with constraint BDDs for specifying block classes using PDs. The following is the basic mapping between SysML and SystemC-AMS:

- SysML Constraint Blocks → SystemC-AMS Modules.
- SysML Flow Ports → SystemC-AMS ports.

Through this concept, we can apply basic map elements between SysML and SystemC-AMS environment specifications and as defined in table1.

Table 1: Planning SysML with SystemC - AMS

Element of SysML	Model of SystemC -AMS
Constraint Block	Module-Core, MoC
Operation	Processes, Event
Requirement	Expression, Assertion
Flow-port	Interface-Channel
Flow-specification	Adaptor
Connector flux	Equation channel
Constraint Property	Equation

Case Study

HD is a medical device used to treat renal failure. Complete renal failure used to be synonymous with death. When a patient's kidneys can no longer do their job, this equipment cleans his blood by passing it through a filter called a dialyzer to remove wastes, excess salt, and water. Blood is directed to the dialyzer using a tube from the patient's arterial access. Blood travels through small fibers inside the dialyzer to filter out waste and surplus fluid. A dialyzate is a chemical agent used in HD to pull fluids and toxins out of the body while supplying electrolytes and other substances to the circulation. After that, the purified blood is reintroduced to the patient via venous access. During HD treatments, vascular access allows huge amounts of blood to circulate constantly in order to refine as much blood as possible. Every minute, a certain amount of blood is pumped through the machine.

The premise of this research is that we will concentrate on WSN energy usage; as a result, we will simplify our case study in order to focus better on our goal. Specify what you want. The following pattern was used to develop the model: Synchronize each requirement with a refinement step.

1. Discriminate and specify the static and dynamic elements of the setting's requirement and the machine of related refinement, respectively.
2. Introduce the safety properties expressed in the requirements as machine invariants.
3. Launch certain monitoring events to show how the HD machine case study requirements can be specified in the SysML model.

In this case, the technical system analyzes and system specifications are as follows:

1. The designed system needs to be economic: This means that the consumption of the energy should be minimized.
2. Emergency drugs and consumables: The system should be available at all times.
3. Every application must be accompanied with documents confirming compliance with local regulations and laws.
4. The system must be safe: The dialysis machine proportioning problems can result in severe serum electrolyte abnormalities. Some of these problems include:

- High or low; serum sodium, potassium, calcium or magnesium.
- High or low; plasma osmolarity due to hyper- r hyperosmolar dialyate.
- Clinical emergencies can occur if significant levels of contaminants are in the dialysate.
- Copper or cupraphane may be released from the heating element or dialyzer and can cause severe hemolysis.
- Chloramines and nitrates can cause severe hemolysis.
- Fluoride can cause severe pruritus, nausea, ventricular tachycardia, or fatal ventricular fibrillation.
- Aluminum can cause bone disease, anemia, and fatal progressive neurologic deterioration, commonly known as dialysis encephalopathy syndrome.
- Lead, copper, zinc, and aluminum can leach from metal pipes and cause anemia.

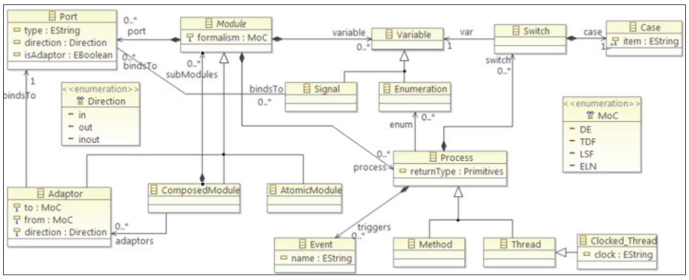

Figure 2: Metamodel of SystemC- AMS Model

We focused on the WSN in our case study to characterize the behavior and operations performed by the sensor node subcomponents to determine parameter values. Receiving, transmission, and data processing are typically performed by the controller and sensor node subcomponents.

The Requirement Diagram

The HD machine's safety requirements are divided into two categories: general requirements (which specify the system's overall behavior) and software requirements (which specify the controller's behavior for each functional area of the system, such as arterial bolus application, blood pump control, and blood-side entry pressure monitoring, for example). Figure 3 depicts the whole set of criteria; however, we will highlight those that are pertinent to our task. When the system disconnects the dialyzer from the dialyzate within 60 seconds and sends an emergency alert in the following scenarios, this is an example of how the HD machine case study criteria are specified:

- The system is in the preparation mode and performs priming or rinsing.
- The system is in the therapy mode, and the dialysate temperature exceeds the maximum temperature of 41C.

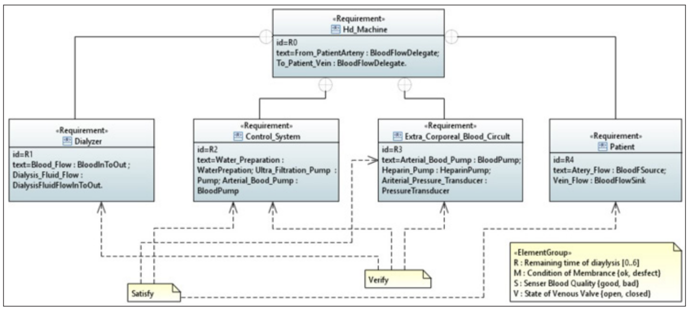

Figure 3: Hemodialysis Machine System Requirement

Designers start by launching a configuration that sets requirement-related static data such modes, operations, and alerts. The refinement's related machine is then given, which includes various variables and invariants. The safety requirements are specified by the accompanying referential integrity of inv1 and inv2.

Then, to record the system's activity, we create two monitoring events, each of which will trigger distinct notifications. When the program is in preparation mode, the event "disconnect Dialyzer

Preparation" is triggered, as illustrated in Figure 3. The temperature of the dialysate rises to over 41C during the operation under these conditions. Since the same thing happens when the application is in treatment mode, the other monitoring event is generated. The "DisconnecitonClock" event dialyzer is also provided to monitor the requirement's timing constraint. The dialyzer Disconnection Time is increased by 1 as the clock ticks (second).

If power is not restored within 15–30 minutes, the system is no longer safe for dialysis, and blood should be manually returned to the patient. If power is not restored within 15–30 minutes, the patient should be removed off the machine. This event is handled to satisfy one of the designed system's most important requirements. Designers can identify two primary requirements for the developed system's safety and durability based on our case study. The monitoring crossing system's requirements diagram is shown in Figure 3.

The Parametric Diagram - Moderate

Model constraints or equations are included into analytic plans using PD, an explicit SysML modeling approach. The parameters and rules that illustrate their progress, which are fundamentally interrelated, determine these restrictions. The main objective of our ongoing research is to investigate WSN energy usage.

Designers used the energy consumption model from to accomplish this. Many sources of energy consumption may be found within the construction of this device, including the transmitter, receiver, sensor, CPU, memory, and actuator [26].

The majority of the software requirements for HD machines are usually timing constraints. We develop patterns for modeling timing constraints such as deadline, expiry, and delay. Instead of using a natural number to represent the global time in the machine, we define a specific timer variable for each timing constraint in order to model the relationship between occurrences of events. We first abstractly present our modeling patterns for the different timing problems before applying them to the HD machine case study. We assume that the national electricity networks are the primary power source supplying the actuator's necessary diagnostic equipment-Blood pressure apparatus, stethoscope, weighing machine, and thermometer. However, we disregard the processor's energy because we have not identified some operations such as collections, compression, or treatments done by the processor. We skipped the energy consumed by the memory since there is no data collected by the sensor node at its level.

The power expended by data transmission, data reception, and the operating mode's transient energy variation, on the other hand, was sustained. Figure 4 shows the energy consumption of this component in a parametric diagram. This diagram represents a transient energy transmitter as well as a communication model that describes the state of the sensor node-system controller transmission channel. As the parametric diagram transmitter, the system defines all parametric diagrams of the processor, memory, receiver, and battery.

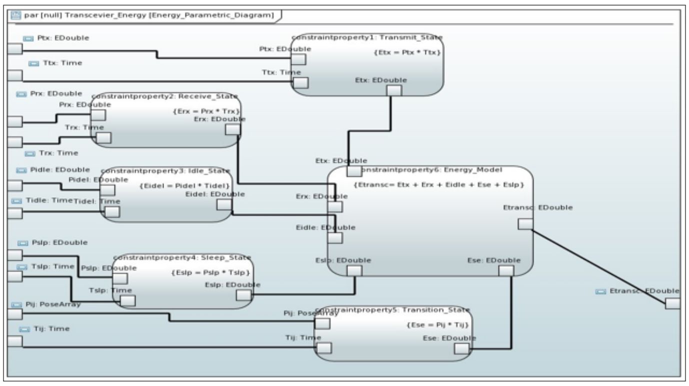

Figure 4: Hemodialysis Machine System Sensor constraint PD

Combine SysML to SystemC- AMS

We are involved with this work on optimizing the system implementation using the best technique, which may be achieved by knowing the parameters that affect the system design. Besides, it is crucial to define the requirement and structure modeling to generate a SystemC AMS environment. We are using the mapping methodology to build the SystemC AMS code from SysML diagrams. Figure 5 demonstrates the SysML2SystemC code.

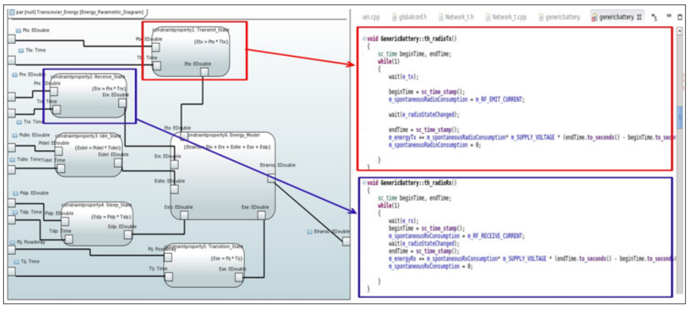

Figure 5: PD to SystemC - AMS Code Creation

The Simulation

Although the code of SystemC is generated successfully from the representation techniques of SysML from the haemodialysis system, the succeeding important step is to simulate the formed SystemC AMS design. This permits low-level modeling, with hardware support, and a simple scalable and tunable architecture. To make an optimum design, we need to have granularity modeling, fine and accurate power consumption analysis, and heterogeneous support. The COSEDA tool Version 2.1 of the simulator proof of concept, has been recently released by COSE DA compliant with the IEEE standard 1666.1 under the Apache 2.0 license [27]. SystemC AMS adds the Timed Data Flow (TDF) model of computation to SystemC, which handles analog signals sampled in time. This late can perform simulations on nodes of heterogeneous sensors that form a network. In this work, we apply the COSE DA tools to perform and demonstrate the behavior of WSN.

The VCD trace format is used in the SystemC standard tracing technique. The parameter activity is recorded in the trace format during communication. The exchange value must be distinct from the thread or method's current value, it is presumed. In addition, the change in this value should last at least a delta cycle. The WSN we have consisted of four nodes and a wireless control channel, with the behavior depicted in Figure 6. A WSN network with a star topology is created using four nodes and one coordinator, and the testbed's simulation yields a VCD trace.

Citation: Abbas A Abdulhameed, Bassam AlKindy, Basil Al-Mahdawi (2022) Smart System Modeling and Simulation design of Hemodialysis Machine by SysML with SystemC-AMS. Journal of Engineering and Applied Sciences Technology. SRC/JEAST-185. DOI: doi.org/10.47363/JEAST/2022(4)153

Using the IEEE 802.15.4 slotted Carrier Sense Multiplexer, all nodes may interact directly with the coordinator. Access with Collision Avoidance (CSMA-CA) algorithm to access the channel. The network nodes are designed to read the environment parameters periodically every second and transmit the data over the network, where each transmission is within two data bytes.

The coordinator and nodes microcontroller are among the essential signals evaluated throughout the simulation process. The radiofrequency unit states Receive, Transmit, Active, and Sleep CooMCUState are among the other key indications evaluated throughout the simulation process. The radiofrequency coordinator country and the coordinator microcontroller country determine these indications. mussitate 0 and radio state 0 for node 0, and mustache 3 and radio state 4 for node 3 are the traditional nodes, microcontroller states, and radiofrequency units.

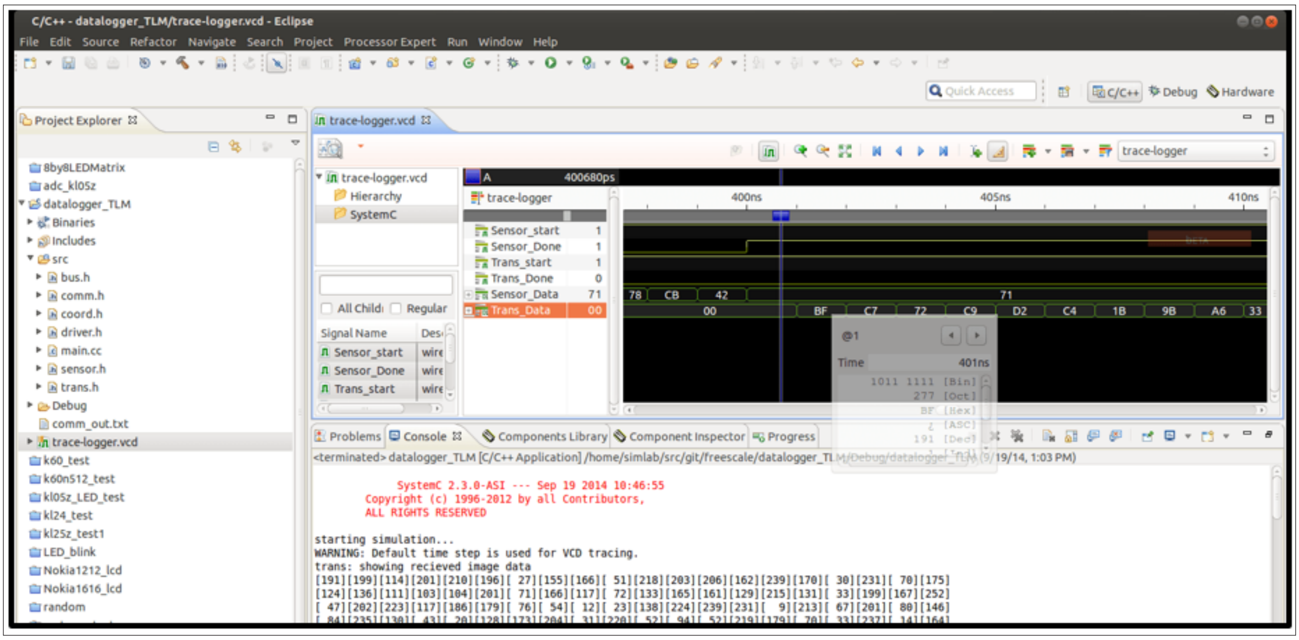

Figure 6: Sensory perception trace created by simulation

Conclusion and Future work

This section presents the simulation platform results that ensure how the packets were routed correctly to their destination under the HD machine.

This paper proposed a novel method to simulate the functional requirements of SysML utilizing constraint block diagrams with parametric diagrams. Being SysML as a semi-formal language, we have presented a recent translation technique of requirement diagrams and constraint block diagrams with a parametric diagram into the SystemC AMS model. This translation was carried out depending on the model conversion performed in the Eclipse platform using Papyrus, ATL, and Acceleo tools. After concluding the SysML requirements, we extracted the properties of these requirements. Afterward, we adopted the COSE DA tool to expand and design our heterogeneous system, where the satisfaction of the extracted properties in the SystemC AMS model is derived from the parametric diagrams. Our verification approach [technique] was guided by "satisfy" and "verify" relationships between requirements, blocks, and parametric diagrams.

By using a case study to describe a Hemodialysis HD system technology, we were able to demonstrate our verification strategy. According to the results, we were able to describe and analyze several important components of HD machines at various abstraction levels using the strategy we used. This method also assisted in detecting and correcting mistakes and omissions that were near to being introduced. In the future, we want to look at the impact of self-control feedback on a system's error detection skills.

Acknowledgment

The authors thanks Mustansiriyah University at (www.uomustansiriyah.edu.iq) and Middle technical University for supporting this study.

Conflict of Interests

The authors declare no conflict of interests.

References

1. Rierson L (2017) Developing safety-critical software: a practical guide for aviation software and DO-178C compliance. CRC Press.
2. Smith S, Kandel A (2018) Verification and validation of rule-based expert systems. CRC Press.
3. Lake Bu, Mark G Karpovsky, Michel A Kinsy (2019) Bulwark: Securing implantable medical devices communication channels. Computers & Security 86: 498-511.
4. Gordon WJ, Stern AD (2019) Challenges and opportunities in software-driven medical devices. Nature biomedical engineering 3: 493-497.
5. Gomes AO, Butterfield A (2016) Modeling the haemodialysis machine with circus. International Conference on Abstract State Machines Alloy B, TLA VDM, Z Springer 409-424.
6. Mashkoor A, Biro M (2015) Towards the trustworthy development of active medical devices: a hemodialysis case study. IEEE Embedded Systems Letters 8: 14-17.
7. Mashkoor A (2016) The hemodialysis machine case study. International Conference on Abstract State Machines Alloy B, TLA, VDM, Z Springer 329-343.
8. Bourouina H, Yahiaoui R, Yusifli E, Benamar MEA, Ghoumid K, et al. (2017) Shear effect on dynamic behavior

of microcantilever beam with manufacturing process defects. Microsystem Technologies 23: 2537-2542.
9. Friedenthal, Sanford, Alan Moore, Rick Steiner (2006) OMG systems modeling language (OMG SysML) tutorial. INCOSE Intl Symp 9: 32.
10. Murti KCS (2022) Specification Languages: SystemC. Design Principles for Embedded Systems. Springer Singapore 85-117.
11. Akaike M, Fukutomi M, Nagamune M, Fujimoto A, Tsuji A, et al. (2012) Simulationbased medical education in clinical skills laboratory. The Journal of Medical Investigation 59: 28-35.
12. Belani, Hrvoje, Marin Vukovic, Željka Car (2019) Requirements engineering challenges in building AI-based complex systems. IEEE 27th International Requirements Engineering Conference Workshops (REW) IEEE.
13. Miorandi, Gabriele, Francesco Stefanni, Federico Fraccaroli, Davide Quaglia (2019) A SystemC-based Simulator for design space exploration of smart wireless systems. Design, Automation & Test in Europe Conference & Exhibition (DATE) IEEE.
14. Guerra A, Esther, Jesús Sánchez Cuadrado, Juan de Lara (2019) Towards effective mutation testing for ATL." 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS) IEEE.
15. Alexandra Mazak, Christine Carpella, Verena Geist, Manuel Wimmer (2020) Thirteen years of SysML: a systematic mapping study. Software and Systems Modeling 19: 111-169.
16. Ichinose M, Sasagawa N, Chiba T, Toyama K, Kayamori Y, et al. (2019) Protein C and protein S deficiencies may be related to survival among hemodialysis patients. BMC nephrology 20: 191.
17. Leupolz J, Habermaier A, Reif W (2018) Quantitative and qualitative safety analysis of a hemodialysis machine with S#. Journal of Software: Evolution and Process 30: e1942.
18. Alenazi, Mounifah, Nan Niu, Juha Savolainen (2019) SysML modeling mistakes and their impacts on requirements. IEEE 27th International Requirements Engineering Conference Workshops (REW) IEEE.
19. Hoang TS, Snook C, Ladenberger L, Butler M (2016) Validating the requirements and design of a hemodialysis machine using iUML-B, BMotion studio, and co-simulation. International Conference on Abstract State Machines Alloy B, TLA, VDM, Z Springer 360-375.
20. Abrial, Jean-Raymond (2018) On B and Event-B: principles, success and challenges. International Conference on Abstract State Machines Alloy B, TLA, VDM, Z Springer Cham.
21. Steve Fotso, Régine Laleau, Hector Barradas, Marc Frappier, Amel Mammar (2019) A Formal Requirements Modeling Approach: Application to Rail Communication. ICSOFT 170-177.
22. Waseem Muhammad, Muhammad Usman Sadiq (2018) Application of model-based systems engineering in small satellite conceptual design-A SysML approach. IEEE Aerospace and Electronic Systems Magazine 33: 24-34.
23. Junger L, Weinstock JH, Leupers R, Ascheid G (2019) Fast SystemC Processor Models with Unicorn," in Proceedings of the Rapid Simulation and Performance Evaluation: Methods and Tools. ACM 2: 1-6.
24. Abdulhameed A, Hammad A, Mountassir H, Tatibouet B (2014) An Approach Combining Simulation¨ and Verification for SysML using SystemC and Uppaal. CAL 8eme conf` erence francophone sur´ les architectures logicielles, Paris, France 9.
25. Abdulhameed A, Hammad A, Mountassir H, Tatibouet B (2015) An approach to verify SysML functional requirements using Promela/SPIN. 12th International Symposium on Programming and Systems (ISPS). IEEE 1-9.
26. Alenazi M, Niu N, Savolainen J (2019) SysML Modeling Mistakes and Their Impacts on Requirements. IEEE 27th International Requirements Engineering Conference Workshops (REW) IEEE 14-23.
27. Porto Rodrigo Cortés, Daniela Genius, Ludovic Apvrille (2019) Modeling and virtual prototyping for embedded systems on mixed-signal multicores. Proceedings of the Rapid Simulation and Performance Evaluation: Methods and Tools 1-7.
28. Daniela Genius, Rodrigo Porto, Ludovic Apvrille, François Pêcheux (2019) A framework for multi-level modeling of analog/mixed signal embedded systems. International Conference on Model-Driven Engineering and Software Development Springer Cham 435-442.
29. Cortés Porto, Rodrigo, Daniela Genius, and Ludovic Apvrille (2021) Handling causality and schedulability when designing and prototyping cyber-physical systems." Software and Systems Modeling 20: 667-683.