Towards Modelling of Cardiac Pacemakers with Timed Coloured Petri Nets and Related Tools[⋆]

Mohammed Assiri and Ryszard Janicki

Department of Computing and Software
McMaster University
Hamilton, Ontario, Canada L8S 4K1
Email: {`assirims, janicki`}`@mcmaster.ca`

Abstract A formal model for the cardiac pacemaker system is presented. Timed Coloured Petri Nets (TCPN) and CPN tools are adopted as modelling tools. The model, which includes various suitable parameters, covers numerous identified characteristics of operation modes and cardiac rhythms in substantial detail. The model can help facilitate a better understanding of the cardiac pacemaker system and provide customizable data to fully evaluate and optimize different algorithms and techniques for the pacemaker system. The obtained results prove the reliability and validity of this model in analyzing the pacemaker system while producing various cardiac events engaging the expressive power and convenience of TCPN.

Keywords: Cardiac Pacemakers · Verification Grand Challenge · Electrocardiogram · Timed Coloured Petri Nets · Biomodelling.

1 Introduction

Verification Grand Challenge is one of the Grand Challenges for Computing Research proposed in [4]. Verification, which is the strict proof of the correctness of software according to its specifications, results in reliable software and potential cost reductions [9]. In addition, the capabilities of other software engineering techniques, such as requirements analysis and testing, can be complemented and extended by verification. Medical devices are well-known examples of safety-critical systems [15] which require significant advances in verification (among other areas). The failure of safety-critical systems not only damages property or the environment but could also lead to human fatality. Therefore, software verification is one approach to prevent such negative consequences.

The cardiac pacemaker (pacemaker thereafter) is an electronic device that monitors and controls the heart rhythm via sensing and pacing operations. The pacemaker treats cardiac arrhythmia, defined as abnormal patterns of the heartbeat. The Software Quality Research Laboratory at McMaster University proposed the pacemaker system specification [5] as a pilot problem for the Verified Software Initiative [27]. This research utilizes formal methods to model and verify the interdisciplinary requirements

---

[⋆] Partially supported by Discovery NSERC Grant of Canada.

of pacemaker systems. It additionally provides customizable data to assess and optimize various algorithms and parameters.

Petri Net-based models have been adopted to model some aspects of pacemakers. In [28], the study's primary purpose was to diagnose cardiac pacemakers' failures remotely. Other studies, such as [18], [19], [21], and [20], focused on runtime verification and pacemaker behaviour rather than the pacemaker system itself.

Moreover, other tools and techniques were applied to model the pacemaker system. The work of [16] used VDM to propose a pragmatic, incremental approach towards modelling the cardiac pacing system. The models were validated through systematic test scenarios, thereby confirming the achievement of the requirements from [5]. In [8], Z notation was used to present a formal specification of a cardiac pacing system where a single state was presented as the base of the overall state of the pacemaker. The outputs were verified and then refined into high-level programming languages. The study of [25] presented an approach that is similar to [8]. Nevertheless, [25] modelled the pacemaker on the fundamental aspect of timing using an algebra-based formalism. Complex behaviours, such as Atrial Tachycardia Response, were uncovered in [25]. Event-B and PROMELA were also employed to model the pacemaker in [22] and [24], respectively. In [22], all operating modes were covered and supportive tools were used for validation. In [24], the SPIN model checking tool was adopted for verification and validation from requirements to implementation. Nevertheless, some advanced modes, such as rate-controlled pacing and hysteresis pacing, were not modelled.

Unlike differential equation-based models or agent-based models, this proposed TCPN-based model does not require a strong knowledge of mathematics or any programming language. The use of TCPNs, which combine graphical notations, hierarchical structures, and supported types of data sets, empowers the representation of the proposed model to be more intuitive and user-friendly as well as formally defined and analyzed.

Furthermore, in this work, not only was the pacemaker system formally verified in accordance with its specifications, the pacemaker system was also integrated into a supplementary model of the cardiac conducting systems to allow for the very detailed modelling of various cardiac rhythms and the adequate validation of the pacemaker. Finally, all processed data from this model is exportable in a customizable fashion for further analysis, optimization, and employment.

2 Background

When a series of electrodes are placed appropriately onto the human body's surface, the cardiac electrical activities can then be recorded as events of signals representing voltage over time. This recording is called the electrocardiogram (ECG) [3]. These ECG events assist in providing meaningful information regarding cardiac arrhythmias and cardiovascular disease.

The ECG is the recording of cumulative signals generated by the cell population at a given time eliciting changes in their membrane potentials. Hence, the ECG does not directly measure the cardiac depolarization and repolarization but rather the changes in the cardiac electrical activities over time [10]. Normal ECG consists of waves, complexes, intervals, and segments following the phases of cardiac conduction. Some signals of ECG patterns indicate the occurrences of specific electrocardiographic activities. Under normal conditions, the most typical events include P wave, QRS complex, T wave, and sometimes U wave, as shown in Figure [10].

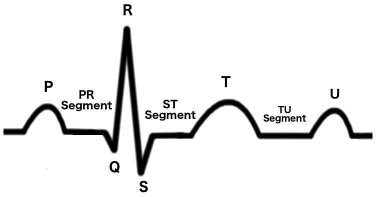

Figure 1: A typical ECG waveform measured from Lead II position

The normal cardiac cycle starts with the stimulation of the senatorial node (SA), located within the right atrium. This initial stimulation is not detected by a typical ECG as the SA is not composed of an adequately large quantity of cells to produce a detectable electrical potential. Instead, the depolarization of SA is conducted throughout the atria, which can be observed as **P wave** in ECG. The action potentials, which next spread throughout the atrioventricular node (VA) and the His bundle, are not large enough to be detectable by ECG. The **QRS complex** indicates the depolarization result of the ventricles. Simultaneously, atrial repolarization effects are masked by this QRS complex due to the immense amount of tissue, thereby causing ventricular depolarization. Thus, atrial repolarization is ordinarily undetectable in ECG. The potential of ventricular repolarization is represented in ECG as **T wave**, and in some individuals, **U wave** represents the late repolarization of papillary muscles. For heart anatomy and the resting membrane potential, the reader is referred to [17].

Some risks associated with heart rhythms can be treated and reduced via the pacemaker system which consists of a pulse generator and one or more leads. The pulse generator (also called the device) is implanted under the skin in the pectoral area (below the collarbone), and it holds a small computer with several electronic circuits and a safely sealed battery within its case. The pulse generator functions include monitoring the heart rhythm, delivering electrical energy (i.e. pulse), and storing information about the heart. Microcomputer-based equipment (called the programmer) communicates with the pulse generator from outside the body via a wand over the skin. The programmer is used to adjust the settings of the pulse generator and retrieve the stored information. The lead is an insulated wire implanted in the heart through veins which then connects to the pulse generator. The lead transfers the heart signal to the pulse generator and returns energy from the pulse generator to the heart to maintain a healthy level of rhythms for the heart. Pacemaker devices are designed to be highly reliable. However, occasional malfunctions may prevent the delivery of therapy. Premature battery depletion, sensing or pacing issues, error codes, and/or loss of telemetry are potential malfunctions of the pacemaker system.

Coloured Petri Nets (CPNs), first proposed in [12] and later substantially modified and enhanced in [14], are an extension of Petri Nets (c.f. [23]) which are often used to model behaviours of rather complex systems. CPNs preserve useful properties of Petri Nets and at the same time extend beyond initial formalism to allow for the distinction between tokens. CPNs is a discrete-event modelling language combining the capabilities of Petri Nets with the capabilities of a high-level programming language. CPNs allow tokens to have a data value attached to them. Such an attached data value is called token colour. Although the colour can be of an arbitrarily complex type, places in CPNs usually contain tokens of one type. This type is referred to as the colour set of the place.

Time in CPNs can be represented in three ways: firing duration, holding duration, and enabling duration [7]. In this proposed model, the holding duration technique is used to classify tokens according to availability, where only available tokens can enable a transition. In [11], the notion of token unavailability is defined implicitly by a timing attribute. Attached to desired tokens, this is called a timestamp, and it is preceded by the symbol @. Time in CPNs is represented as simulated time: a symbolic representation of time. Simulated time and real physical time have no intrinsic relationship whatsoever, and a built symbolic representation of real-time timestamps belongs to Time Set (TS), which is equal to $\mathbb{R}_{\geqslant 0}$. The timed markings are multi-sets on TS, where TS_{MS} is denoted by a collection of timestamps.

A semi-formal definition, sufficient for our purposes, can be given as follows [11]: A Coloured Petri Net is a tuple:

$$CPN = (P, T, A, \Sigma, C, N, E, G, I)$$

where:

- P is a set of places, T is a set of transitions, A is a set of arcs.
- $P \cap T = P \cap A = T \cap A = \emptyset$
- Σ is a set of colours.
- C is a colour function that maps places in P into colours in Σ.
- N is a node function that maps A into $(P \times T) \cup (T \times P)$.
- E is an arc expression function that maps each arc $a \in A$ into the expression e. The input and output types of the arc expressions must correspond to the type of nodes that the arc is connected to.
- G is a guard function. It maps each transition $t \in T$ into guard expression g. The output of the guard expression must be evaluated as true or false.
- I is an initialization function. It maps each place p into an initialization expression i. The initialization expression must evaluate a multi-set of tokens with a colour corresponding to the colour of the place $C(p)$.

The Timed Coloured Petri Net or TCPN is defined as follows [6]:

$$TCPN = (CPN, f, M, M_0)$$

where:

- $CPN = (P, T, A, \Sigma, C, N, E, G, I)$ is a Coloured Petri Net.
- $f : T \rightarrow [0, \infty)$ is a time transition function.

- $M : P \rightarrow [0,\infty)$ is a timed marking and M_0 is the initial marking.

In this paper, $\mathbb{R}$ is used to denote real numbers and $\mathbb{Z}$ is used to denote integer numbers.

There are a variety of tools that can be used for CPNs (c.f. [13, 26]). In this paper, the **CPN Tools v.4.0** from [2] has been used. The CPN Tools is composed of a graphical editor for constructing models, a State Space tool for verifying properties of models, and a simulator for executing the net. For more details and theory of CPN, the reader is referred to [11].

3 System Requirements

The pacemaker system is a hybrid embedded real-time system; its development requirements are given and organized in [5]. Due to the lack of space, the work of this paper is limited to the following requirements:

- The pacemaker shall support single and dual chamber rate-adaptive pacing.
- The device shall output pulses with programmable voltages and widths (atrial and ventricular) which provide electrical stimulation to the heart for pacing.
- The atrial and ventricular pacing pulse amplitudes shall be independently programmable.
- The atrial and ventricular pacing pulse widths shall be independently programmable.
- Rate sensing shall be accomplished using bipolar electrodes and sensing circuits. All rate detection decisions shall be based on the measured cardiac cycle lengths of the sensed rhythm. Rate shall be evaluated on an interval-by-interval basis.
- The following bradycardia operating modes shall be programmable: Off, DDDR, VDDR, DDIR, DOOR, VOOR, AOOR, VVIR, AAIR, DDD, VDD, DDI, DOO, VOO, AOO, VVI, AAI, VVT and AAT. As well, OVO, OAO, ODO, and OOO shall be available in temporary operation. (see Table 1).

Pacemaker Code The pacemaker's operating modes are identified using a *Pacemaker Code*, which is a sequence of four valid letters representing four categories. Table 1 shows the four categories and their valid letters. For example, the mode *AOO* indicates that the pacemaker paces the atria without sensing any chamber. *AAI* mode indicates that the pacemaker paces and senses the atria and releases a pulse when there is no natural pulse. Groping operating codes are also possible by using the letter *X* as *wildcard notation* to denote any letter. For example, the mode *AXX* represents all modes that start with an *A*.

4 TCPN-based Modelling of Cardiac Pacemakers

Different from former studies, this proposed model is composed of seven interconnected but independent submodels. Such a structure not only empowers design simplicity and flexibility but also improves observability and design coupling. These submodels include the **core-elements** submodel, the **atrial-depolarization** submodel, the **AV-node** submodel, the **ventricular-depolarization** submodel, the **ST-segment** submodel,

Table 1: The operation mode of the pacemaker

I	II	III	IV
Chamber(s) paced	Chamber(s) sensed	Response to sensing	Rate modulation
O = None	O = None	O = None	R = Rate modulation
A = Atria	A = Atria	T = Triggered	
V = Ventricle	V = Ventricle	I = Inhibited	
D = Dual (A+V)	D = Dual (A+V)	D = Traced	

the **ventricular-repolarization** submodel, and the artificial-pacemaker submodel. The functions and connections between these submodels are described as follows: The core-elements submodel represents the common four places among other submodels. This submodel is a part of all other submodels that exchange data across it. The atrial- depolarization submodel captures the activities of the depolarization of atrial musculature. In a healthy heart, the cardiac cycle begins with the firing of the SA node (i.e. the natural pacemaker) followed by the depolarization of atrial musculature producing the recordable P-wave in the ECG. The AV-node submodel addresses the event that follows the end of atrial depolarization. When action potentials spread through the AV node, this results in the PR segment in ECG. This is the flat line between the end of the P wave and the start of the QRS complex. The ventricular-depolarization submodel processes the right and left ventricles' depolarization and the recordable QRS complex. The ST segment submodel addresses the time interval between the ventricular depolarization and repolarization, called the ST segment in ECG. The ventricular-repolarization submodel, as the name suggests, presents ventricular repolarization. Eventually, the artificial-pacemaker submodel monitors all the cardiac activities, ensures the implementation of pacemaker rules, and fires when specific rules, set by the programmed parameters, are met.

4.1 The Core-Elements Submodel

This submodel, as shown in Figure 2, consists of four places: ⟨*Pacemaker parameters*⟩, ⟨*ECG Events*⟩, ⟨ *Intervals and Ratios*⟩ and ⟨*Timing Cycle*⟩. These places also belong to other submodels. The main reason for modelling the core-elements as an independent submodel is that all other submodels contain these places of the core-element submodel. Therefore, the places are defined in an independent submodel that has a recognizable name.

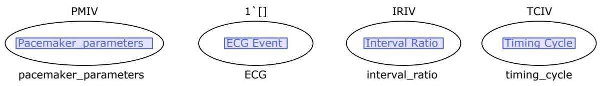

Figure 2: The Core-Elements Submodel

The first place has the colour set **Pacemaker Parameters**, which is a record colour set or the Cartesian product of the sets as described in Table 2.

Table 2: The definitions of colour set **Pacemaker Parameters**

Colour Set	Definition
pacing source	*SA node* $\cup$ *AV node* $\cup$ *Purkinje Fibers* $\cup$ *None* $\cup$ *Artificial Pacemaker*
rate condition	*Fibrillation* $\cup$ *Flutter* $\cup$ *Tachycardia* $\cup$ *Normal* $\cup$ *Bradycardia* $\cup$ $\{$*NA*$\}$
code symbols	$O \cup D \cup A \cup V \cup T \cup I \cup R$
operation mode	$\{$(*I, II, III, IV*) $\mid$ *operation mode* $\in$ *code symbols* $\}$
pacemaker Parameters	$\{$(*mode, last paced chamber, last sensed chamber, source, assigned bpm, rate condition, Pwave-based bpm, Pwave rate condition, Rwave-based bpm, Rwave rate condition, heartbeats counter*) $\mid$ *mode* $\in$ *operation mode, last paced chamber, last sensed chamber* $\in$ *code symbols, source* $\in$ *pacing source, assigned bpm, Pwave-based bpm, Rwave-based bpm, heartbeats counter* $\mathbb{Z}_{\geqslant 0}$, *rate condition, Pwave rate condition, Rwave rate condition* $\in$ *rate condition* $\}$

In the colour set **Pacemaker Parameters**, [*mode*$\rangle$ refers to the operation mode of the artificial pacemaker as explained in Table 1. In dual-chamber modes, the $\langle$*last paced chamber*$\rangle$ and $\langle$*last sensed chamber*$\rangle$ are utilized to maintain the order and apply the rules accordingly. Also, $\langle$*source*$\rangle$ indicates the generator of a heartbeat. In a healthy heart, the Sinoatrial node (SA) is the natural pacemaker as it has the most rapid firing. Nevertheless, other cells can fire spontaneously at a slower rate as subsidiary pacemakers if the SA node permanently or temporarily ceases to fire. Consequently, $\langle$*source*$\rangle$ is determined based on the value of the parameter $\langle$*set bpm*$\rangle$ as defined in Table 6, and it is controlled by means of guard function on selected transitions. Similarly, $\langle$*rate condition*$\rangle$ is based the value of parameter $\langle$*set bpm*$\rangle$ as demonstrated in Table 3.

Table 3: The conditions of the heart rate for adults

Rate Condition	Fibrillation	Flutter	Tachycardia	Normal	Bradycardia
Value of *set bpm*	> 350	$251 - 350$	$101 - 250$	$60 - 100$	< 60

In the colour set **Pacemaker Parameters**, $\langle$*heartbeat counter*$\rangle$ traces the number of generated full heartbeats. $\langle$*Pwave-based bpm*$\rangle$ and $\langle$*Rwave-based bpm*$\rangle$ refer to the heart rate based on the calculation of PP-interval and RR-interval. In ECG, one method to calculate the heart rate is to measure the atrial rate, indicated by PP-interval, or the ventricular rate, indicated by RR-interval. PP-interval and RR-interval represent the time between successive P waves and R waves, respectively.

The second place, as shown in Figure 2, has the colour set **ECG**. In principle, this is a list of generated synthetic information from **ECG Event**, which are defined as a record colour set or the Cartesian product of the sets described in Table 4. By default, each event, when it occurs, inserts a single element of the set **ECG Event** into the list of the place $\langle$*ECG*$\rangle$. However, the events of fluctuated waves insert a number of elements, with proper calculation of cumulative time, into the list of *ECG* based on the value of the parameters $\langle$*Fx*$\rangle$ and $\langle$*fx*$\rangle$ demonstrated in Table 6. Upon generating a new event, the content of the list of **ECG** is dropped, and then the new event is inserted to avoid

Table 4: The definition of colour set **ECG**

Colour Set	Definition
event	{*P-wave, Q-wave, R-wave, S-wave, T-wave, U-wave, PR-segment, ST-segment, TU-segment, TP-segment, F-wave, f-wave, dropped-Pwave, dropped-Qwave, dropped-Rwave, dropped-Swave, dropped-Twave, dropped-Uwave, dropped-PRsegment, dropped-STsegment, dropped-TUsegment,paced-Pwave, paced-Qwave, paced-Rwave, paced-Swave, paced-Twave, paced-Uwave, paced-PRsegment, paced-STsegment, paced-TUsegment, paced-TPsegment,artificial-pacing, artificial-sensing, atrial-pacing, ventricular-pacing*}
bipolar limb leads	{(*I, II, III*) | *bipolar limb leads* $\in \mathbb{R}$}
augmented limb leads	{(*aVR, aVL, aVF*) | *augmented limb leads* $\in \mathbb{R}$}
precordial leads	{(*V1, V2, V3, V4, V5, V6*) | *precordial leads* $\in \mathbb{R}$}
leads (i.e. the standard 12 leads)	*bipolar limb leads* $\cup$ *augmented limb leads* $\cup$ *precordial leads*
ECG Event	{(*ID, title, start time, end time, duration, amplitude*) | *ID* $\in \mathbb{Z}_{\geqslant 1}$, *title* $\in$ *event, start time, end time, duration* $\in \mathbb{R}_{\geqslant 0}$, *amplitude* $\in$ *leads*}

overflow. Nevertheless, all events can be exported into text files through a monitoring tool within CPN Tools, as discussed in section simulation-based analysis.

As demonstrated in Table 4, each token of **ECG Event** has a unique ⟨*ID*⟩ and ⟨*title*⟩ for the purpose of identification. The ⟨*start time*⟩ of the event equals the ⟨*end time*⟩ of the previous event, with the exception of the initial event where the ⟨*start time*⟩ equals the value of the parameter ⟨*initial time*⟩ as defined in Table 6. The ⟨*end time*⟩ of the **ECG event** is calculated as follows:

$$calculated\ duration \approx U(\omega - \kappa, \omega + \kappa) \tag{1}$$

$$end\ time = calculatedduration + start\ time \tag{2}$$

where:

- U stands for uniform distribution.
- ω is the value of the duration of the parameter ⟨*Event Duration*⟩.
- κ is the value of the range of the parameter ⟨*Event Duration*⟩.

Hence, the ⟨*calculated duration*⟩ of an ECG event can have either a constant value in every heartbeat, or it can be assigned continuously via the uniform distribution calculation for an arbitrary value w.r.t. assigned parameters.

As ECG events are typically investigated using a number of different electrode positions or configurations, the ⟨*amplitude*⟩ of the colour set **leads** (defined in Table 4) denotes the estimated total height of events through the values of 12 standard ECG leads serving 12 different perspectives. The value of ⟨*amplitude*⟩ is supplied by the parameter ⟨*Event Amplitude*⟩. The connection of leads is presented by the parameters ⟨*Bipolar Limb Leads*⟩, ⟨*Augmented Limb Leads*⟩, and ⟨*Precordial Leads*⟩ as defined in Table 6. If the corresponding lead parameter is set to *false*, then the amplitude of that lead is zero, which theoretically states an inactive lead. Otherwise, the amplitude of the lead is calculated as previously discussed.

The third presented place in Figure 2 is ⟨*interval ratio*⟩, and it has the colour set **Interval Ratio** defined in Table 5. In principle, it is a set of numerical trackers that support calculate the intervals and ratios of ECG events. The PP-interval and RR-interval are parts of the heart rate calculation previously explained in colour set **Pacemaker Parameters**. The event ratio is a pair:

$$(\eta, \mu)$$

where:

- η is the frequency of dropping the event (w.r.t. ⟨*Event Duration*⟩)
- μ is the total number of occurrences before the event is dropped.

Table 5: The definition of colour set **Interval Ratio**

Colour Set	Definition
interval ratio	$\{($*P previous interval, P last interval, R previous interval, R last interval, heartbeat counter, heartbeat ratio, P ratio, Q ratio, R ratio, S ratio, T ratio, U ratio, PR ratio, ST ratio, TU ratio*$) \mid$ *interval ratio* $\in \mathbb{Z}_{\geqslant 0}\ \}$

The event ratio is ruled by two parameters, namely ⟨*Event Enablement*⟩ and ⟨*Event Ratio*⟩ as defined in Table 6. The main distinction between these two parameters is their consideration of the event's duration time. If an event's enablement is set to false, then it will disappear from the ECG events. However, if the event is enabled and its ratio is fulfilled, then the event is substituted by the corresponding set of ⟨*dropped event*⟩, which holds the event's assigned duration with zero amplitudes. Similarly, the parameters ⟨*Dropped Beat Enablement*⟩, ⟨*Dropped Beat Ratio*⟩ and ⟨*Dropped Beat Duration*⟩ respectively control the occurrence, frequency, and duration for the case of dropped beats during the cardiac conduction cycle.

The ⟨*Timing Cycle*⟩ place in Figure 2 has the colour set **Timing Cycle**, which is a record colour set or the Cartesian product of the sets as defined in Table 7. The ⟨*ID*⟩ in the colour set **Timing Cycle** increases by one each time the artificial pacemaker is sensing or pacing following the defined rules (see artificial pacemaker submodel below). The ⟨*next event*⟩ and ⟨*last event*⟩ are utilized for accurate concurrent operations between the pacemaker and the heart activities if the assigned pacemaker's operation mode declares certain engagements. As some of the pacemaker's operation modes, such as in *AOO*, can pace a chamber while the heart is conducting another operation, the ⟨*event-chunked*⟩, ⟨*ecg-duration*⟩ and ⟨*remain-duration*⟩ in the colour set **Timing Cycle** control the computing of the concurrent events and, when applicable, distinguish paced events as defined in Table 4. The ⟨*pacing-rate*⟩ is calculated based on the assigned operation mode. In most modes, the ⟨*Low Rate Limit (LRL)*⟩ is the pacing rate, but in other modes where the ⟨*Rate modulation*⟩ (i.e. XXXR) is set, the ⟨*Maximum Sensor Rate (MSR)*⟩ is considered during calculation. The values of ⟨*current-pacing-occurrenc*⟩ and ⟨*next-pacing-duration*⟩ are continuously computed with every new cycle based on the programmed parameters. Finally, the colour set of ⟨*timers*⟩ represents the computed values of the assigned parameters explained in Table 6.

Table 6: Parameters

Parameter	*Value*	*Explanation*
operation mode	Off DDD VDD DDI DOO AOO AAI VOO VVI AAT VVT DDDR VDDR DDIR DOOR AOOR AAIR VOOR VVIR	As defined in Table 2
Timers	$\mathbb{R}_{\geqslant 0}$	As defined in Table 7
LRL	$30-175$	The minimum number of beats per minute
URL	$50-175$	The maximum rate at which the paced ventricular rate will track sensed atrial events
AV Delay	$70-300$	The programmable time period from an atrial event to a ventricular pace
Dynamic AV Delay	{*true, false*}	Controls the individual AV delay for each cycle
Sensed AV Delay Offset	$-100-0$	Shortens the AV delay following a tracked atrial sense
A amplitude	$\mathbb{R}_{\geqslant 0}$	Atrial Pulse Amplitude
V amplitude	$\mathbb{R}_{\geqslant 0}$	Ventricular Pulse Amplitude
A width	$\mathbb{R}_{\geqslant 0}$	Atrial Pulse Width
V width	$\mathbb{R}_{\geqslant 0}$	Ventricular Pulse Width
VRP	$150-500$	Ventricular Refractory Period
ARP	$150-500$	Atrial Refractory Period
PVARP	$150-500$	Post Ventricular Atrial Refractory Period
PVARP Extension	$0-400$	Extend PVARP for certian cases
ATR mode	{*true, false*}	Atrial Tachycardia Response duration
ATR duration	$10-200$	Atrial Tachycardia Response
ATR Fallback	$1-5$	Atrial Tachycardia Response Fallback
MSR	$50-175$	Maximum Sensor Rate
Activity Threshold	V-Low, Low, Med-Low, Med, Med-High, High, V-High	The value the accelerometer sensor output shall exceed
Response Factor	$1-16$	The pacing rate that occurs at various levels of steady state patient activity
Reaction Time	$10-50$	The rate of increase of the pacing rate
Recovery Time	$2-16$	The rate of decrease of the pacing rate
Rate Smoothing	$0-25$	The pacing rate change that occurs due to precipitous changes in the intrinsic rate
initial time	$\mathbb{R}_{\geqslant 0}$	The initial start time of the model
number of beats	$\mathbb{Z}_{\geqslant 0}$	The maximum number of simulated heartbeats. If the input value equals zero, then the number of simulated heartbeats will be unrestricted.
set bpm	$\mathbb{Z}_{\geqslant 0}$	beats/min
Event Enablement	{*true, false*}	For all events defined in Table 4
Event Duration	$(duration, range)$ where $duration, range \in \mathbb{R}_{\geqslant 0}$	As explained in equation 1
Fluctuated Wave (i.e. *Fx or fx*)	$\mathbb{Z}_{\geqslant 0}$	The repetition number of the wave per single beat.
Event Amplitude	$\mathbb{R}$	For all leads defined in Table 4
Event Ratio	$\mathbb{Z}_{\geqslant 0}$	For elements defined in Table 6. If the input value equals zero, then the event will always be represented (i.e. never dropped).
Dropped Beat Enablement	{*true, false*}	c.f. Dropped Beat Ratio
Dropped Beat Ratio	$\mathbb{Z}_{\geqslant 0}$	beat:beat(s) as explained in equation 1
Dropped Beat Duration	$\mathbb{R}_{\geqslant 0}$	If the input value equals zero, then the duration of one dropped heartbeat automatically equals the total time of all enabled events
Connections of Leads	{*true, false*}	For all leads defined in Table 4

Table 7: The definition of colour set **Timing Cycle**

Colour Set	Definition
timers	$\{(\textit{ARP} \cup \textit{VRP} \cup \textit{LRL} \cup \textit{URL} \cup \textit{MSR} \cup \textit{ATR} \cup \textit{AV-delay} \cup \textit{V-blanking} \cup \textit{A-blanking} \cup \textit{PVARP} \cup \textit{VA-interval}) \mid \textit{timers} \in \mathbb{R}\ \}$
timing cycle	$\{(\textit{ID, next event, last event, event-chunked, pacing-rate, ecg-duration, remain-duration, time-tracker, current-pacing-occurrenc, next-pacing-duration, timers}\ ) \mid \textit{ID, event-chunked} \in \mathbb{Z}_{\geqslant 1}$ *next event, last event* $\in$ *event, pacing-rate, ecg-duration, remain-duration, time-tracker, current-pacing-occurrenc, next-pacing-duration* $\in \mathbb{R}_{\geqslant 0}$, *timers* $\in$ *timers*$\}$

Table 6 defines and explains some parameters for the proposed model. Modelling the pacemaker involves conducting its best simulation based on the selected values of the parameters. More accurate values of the parameters lead to more reliable results generated by the model.

4.2 The Atrial-Depolarization Submodel

As shown in Figure 3, this submodel demonstrates the recorded ECG event, P-wave, during the atrial depolarization. The processing of P-wave, if enabled, is determined based on the computing of the tokens from all the four connected places explained in the previous submodel. The token of place ⟨*Timing Cycle*⟩ has the timestamps of a real number, which initially equals the value of the parameter ⟨*initial time*⟩, and after that equals the continuous cumulative timestamps of the model. The firing of transition ⟨*stimulate atrial*⟩ is only enabled if its guard, which represents appropriate rules, is holding. The specific rules that must be satisfied are as follows:

- The value of ⟨*next event*⟩ from the colour set **Timing Cycle**, defined in Table 7, equals P-wave, dropped-Pwave, or paced-Pwave.
- If the value of the parameter ⟨*number of beats*⟩ is greater than zero, then the number of modelled full heartbeats must be less than or equal to the value of this parameter.
- If the value of the parameter ⟨*number of beats*⟩ equals zero, then the number of modelled full heartbeats is not restricted.

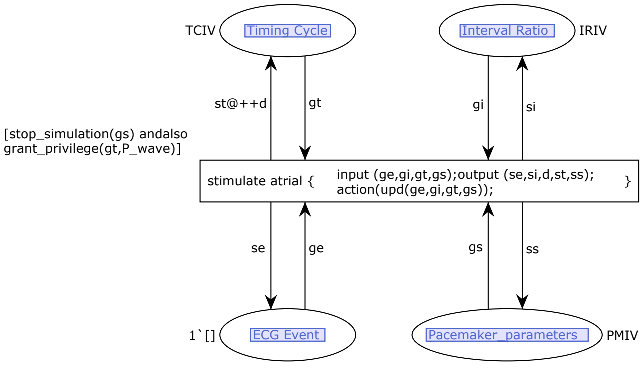

Figure 3: The Atrial-Depolarization Submodel

After firing transition ⟨*stimulate atrial*⟩, all tokens are computed according to the event's parameters, and they are returned to their original places.

4.3 The AV-Node Submodel

Like the previous submodel, the AV-Node submodel (Figure 4) represents the ECG event's PR segment. The transition ⟨*present PR segment*⟩ is enabled if and only if the value of ⟨*next event*⟩ from the colour set **Timing Cycle** equals PR-segment, dropped-PRsegment, or paced-PRsegment. Upon the firing of transition ⟨*present PR segment*⟩, all tokens from all connected places are concurrently calculated by their defined parameters and then returned to their original places.

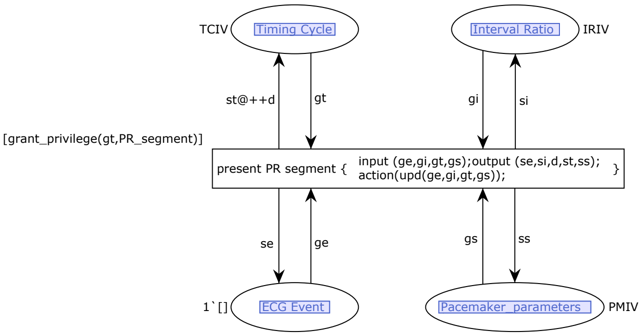

Figure 4: The VA-Node Submodel

4.4 The Ventricular-Depolarization Submodel

The Ventricular-Depolarization submodel, as shown in Figure 5, discusses the QRS complex, which expresses the ventricular depolarization. This submodel is composed of three transitions interpreting the waves of the QRS complex. Each transition is enabled and fired when its guard is fulfilled according to the value of ⟨*next event*⟩ from the colour set **Pacemaker Parameters**. Consequently, the tokens of connected places are appropriately updated.

4.5 The ST-Segment Submodel

In Figure 6, the ST-segment submodel, as the name suggests, denotes the ST segment of the ECG event. When transition ⟨*present ST segment*⟩ is fired, tokens are computed as in previous submodels.

4.6 The Ventricular-Repolarization Submodel

This submodel demonstrates the repolarization of ventricular through the ECG events; T-wave, TU-segment, U-wave, and TP-segment. As the tokens are computed via firing fulfilled transitions, those tokens of associated places are modernized thoroughly concerning the related parameters' values as discussed in the core-elements submodel.

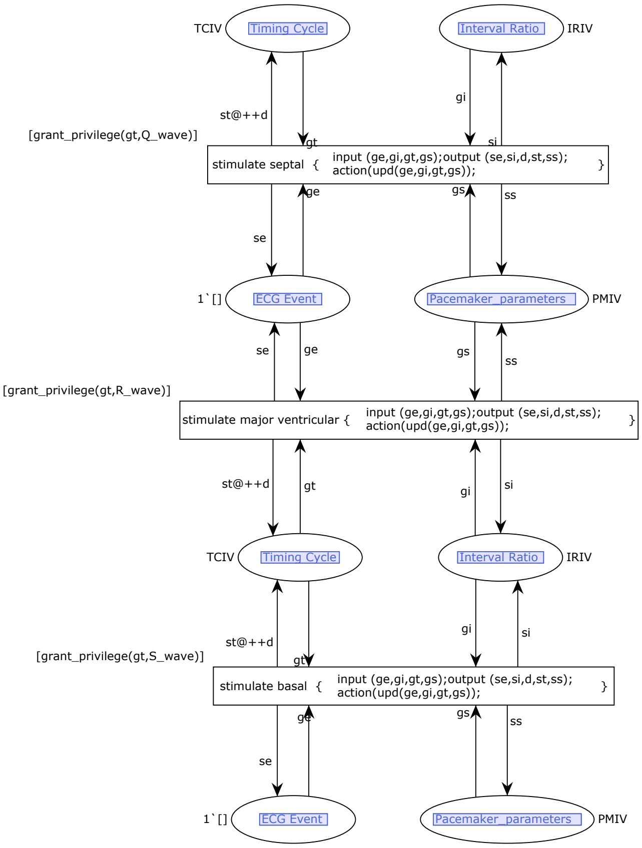

Figure 5: The Ventricular-Depolarization Submodel

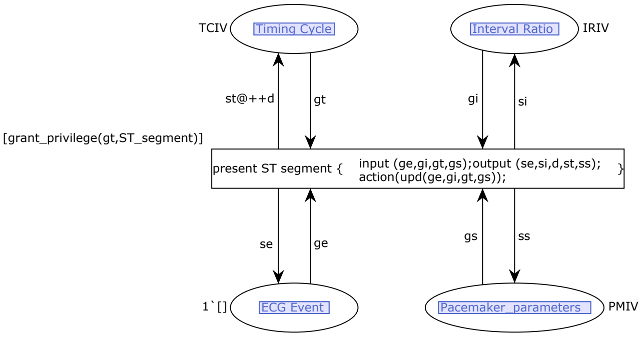

Figure 6: The ST-Segment Submodel

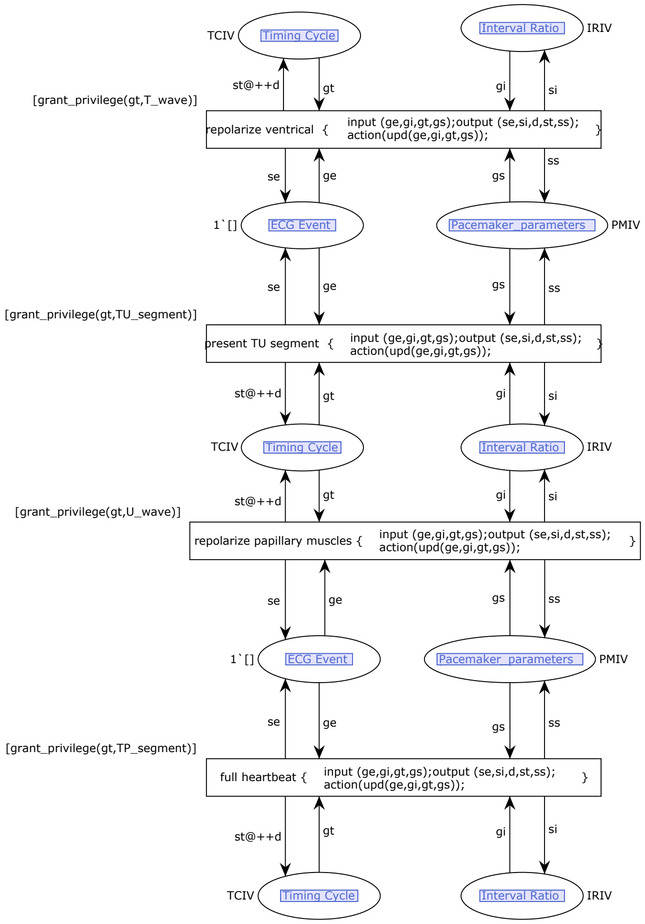

Figure 7: The Ventricular-Repolarization Submodel

4.7 The Artificial-Pacemaker Submodel

This last submodel (Figure 8) signifies the artificial pacemaker's functions and operations. The firing of transition ⟨*artificial pacing*⟩ is only enabled if its guard, which considers relevant rules, is holding. The specific rules that must be satisfied are as follows:

- The value of parameter ⟨*operation mode*⟩ (Table 6) indicates chamber pacing as in *AXXX, VXXX,* and *DXXX* (Table 1).
- The value of ⟨*next event*⟩ from the colour set **Timing Cycle**, defined in Table 7, equals ⟨*artifical-pacing*⟩.
- The current time of the model equals the calculation of ⟨*pacing rate*⟩ (Table 7)
- The associated ⟨*timers*⟩ of the applied ⟨*operation mode*⟩ equal their assigned values.

After firing transition ⟨*artificial pacing*⟩, tokens are computed accordingly, and then associated places are updated with the processed tokens.

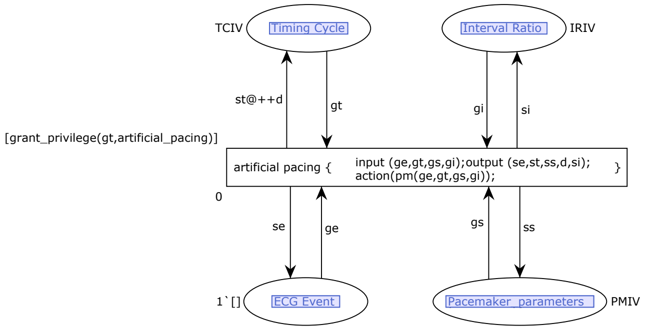

Figure 8: The Artificial-Pacemaker Submodel

5 The Results of Analysis

The analysis is conducted utilizing a test scenario methodology. All the required operation modes, declared by the requirements, were exhaustively verified in accordance with their associated parameters. These operation modes were also validated with various cardiac rhythms where assumed results were sustainably met. For instance, one of the analyzed test scenarios is a natural pacemaker (i.e. the heart) that failed with combined sinus node dysfunction and AV node dysfunction. Usually, a dual-chamber pacemaker with operation mode DDD or DDDR is chosen. Subsequently, the values of the associated parameter of the selected operation mode were configured following the system specifications [5]. Then, the proposed model was verified and validated thoroughly via the employed techniques.

Various techniques and tools can be applied to analyze CPN-based models [1]. In this work, two techniques were practiced, namely the simulation-based performance analysis and the State Space Analysis (the reachability analysis). The simulation-based performance analysis is based on continually simulating the model while its data is being monitored and recorded and then measured and compared. Although this technique is flexible, it is also labour-intensive. The reachability analysis is a formal verification technique typically executed by a computer tool to construct a graph of nodes rendering reachable markings and arcs connecting enabled transitions of the markings. The State Space tool [13] was applied to analyze the proposed model and generate a specific property report.

5.1 The Simulation-based Analysis

The simulation of a CPN-based model can be expressed as the process of firing a sequence of enabled transitions. The simulation-based performance analysis is based on continually simulating the model while its data is being monitored, recorded, and then measured and compared. This technique was applied by the CPN Tools simulator tool along with monitors supported by the CPN Tools. The monitors are data-collectors and controller techniques that guide the simulation of the model. The model was analyzed through interactive and automatic simulations with different parameters' values (Table 6).

The significant results of the simulation-based analysis include the following:

- A more solid understanding of and reliance upon the correctness of the model's design was gained. Simulation modelling assists in recognizing and debugging errors in the model's constructions and definitions.
- The comprehensiveness of the model's specification was assured. This is important as incomplete specifications prohibit the model's desired execution.
- Design flaws were corrected and particular sequences were examined to reflect the system's behaviour in selected scenarios.
- The simulations show that the model appears to terminate in the desired consistent state under its parameters correctly.
- In addition, user-defined monitors were applied to selectively export generated tokens of the ECG events into text-based files. Such exported data can then be utilized to adequately evaluate and optimize different algorithms and techniques.

Nevertheless, the simulation-based technique cannot guarantee that all possible executions are covered; therefore, State Space analysis is conducted to verify other functional and performance properties.

5.2 State Space Analysis

State Space analysis is a formal verification technique typically performed by a computer tool to construct a directed graph of arcs and nodes. The nodes represent reachable markings (the model's states), and the arcs represent enabled binding elements (state transformations). As enabled binding elements are executed until all reachable markings are generated, thereby giving all possible model states, various properties of the model can then be verified. The behavioural properties reported by the Space State analysis include some statistical information, boundedness properties, and other properties.

The strongly-connected-component graph (SCC graph) is often derived from the State Space's graph structure. CPN State Space tool uses the SCC graph to assess the standard behavioural properties of the model. Nevertheless, the calculation of timed state space can be complicated and laborious in part because the size of the reachability graph can be infinite as several timed markings with global clock and timestamps are distinguished. In such a case, the model can be analyzed for a defined period of time to limit the infinite markings due to the global clock.

In this paper, four scenarios are analyzed. Each scenario has a specific operation mode and parameters set to the norm values. These selected operation modes are *OAO, DOO, VDD*, and *DDDR*, with a total of 1000, 2000, 3000, and 6500 heartbeats, respectively.

For all operation modes, the reported statistical results hold similar features. As shown in Table 8, the numbers of nodes and arcs in the SCC graph are always identical to State Space's corresponding numbers. As expected, this implies that the model has no cyclical behaviour. Even though the numbers of nodes and arcs are equal for both State Space and SCC graph in this proposed model, these numbers may not always agree based on the model's specifications. When the number of SCC-graph nodes is fewer than the number of State-Space nodes, this indicates that there are non-trivial SCCs and cycles in the State Space of the model (i.e. may not terminate).

Table 8: CPN Tools state space report

Statistics					
Operation Mode		OAO	DOO	VDD	DDDR
Number of beats		1000	2000	3000	6500
Occurrence Graph (State Space)	Nodes	1001	2005	3001	6505
	Arcs	1000	2004	3000	6504
	Status	Full	Full	Full	Full
Scc Graph	Nodes	1001	2005	3001	6505
	Arcs	1000	2004	3000	6504
Dead Markings		[1001]	[2005]	[3001]	[6505]
Dead Transition Instances		[1][a]	None	[1][b]	None
Live Transition Instances		None	None	None	None

---

[a] Artificial Pacemaker(artificial pacing)

[b] Artificial Pacemaker(artificial pacing)

The reachability properties' results indicate that an occurrence sequence exists in a sequential direction. By performing standard query functions, it was observed that the occurrence sequence, in all scenarios, returns true only between nodes in ascending order. This behaviour is anticipated since the model is time-based. The boundedness properties report the number of tokens for places after executing all reachable markings. In all scenarios, the quantity of one token was reported as the maximal and minimal number of tokens that can remain, in any reachable marking, on each place. Similar to reachability properties' results, the boundedness properties' results matched the assumed numbers in accordance with the timed characteristics of this model. The home properties' results, by design, indicate no home marking that is reachable from any other marking.

The liveness properties' results address *dead markings*, *dead transition instances*, and *live transition instances*. *Dead markings* are those with unenabled binding elements which result from nodes holding tokens with no outgoing arcs or disabled transitions. For all scenarios, a single dead marking was reported. This is typically expected with models that should terminate at a certain point by design. The results of *dead transition instances* indicate the transitions that were never enabled (i.e. fired) during the model's execution. When the results of *dead transition instances* are reported as *None*, this implies that all the transitions in the model have the opportunity of firing (occurring) at least once. According to the adopted operation modes and parameters, two scenarios hold no dead transition instances. In comparison, each of the other two scenarios register a single dead transition instance. This result is interpreted as indicating that the pacemaker did not deliver any electrical impulses following the designated parameters. On the other hand, the results of *live transition instances* state the transitions that are reachable in the occurrence sequence of any reachable marking. All four scenarios contain no *live transition instances*, thereby determining that each transition is not always found in the occurrence sequence of any reachable marking. Models with dead markings regularly have no *live transition instances* because the model was terminated. Dead transitions are not the opposite of live transitions since a non-dead transition must be enabled at least once while a live transition should continue to be enabled. Finally, the fairness properties' results represent a list of all impartial transitions which occur infinitely. When such transitions are removed or restricted, all infinite occurrence sequences of the model will be subsequently eliminated if desired. As expected in accordance with the parameters and specifications, there are no infinite occurrence sequences in any scenario.

6 Conclusion

Software verification is an ongoing challenge. More studies and efforts in this area will eventually assist in the development of more reliable software. As a pilot problem for the Verified Software Initiative, modelling the pacemaker system through different formalisms will not only lead to a more concrete understanding of current tools but will also drive more comprehensive comparison among them. This paper demonstrates the expressive power and reliability of TCPNs towards modelling safety-critical systems such as the pacemaker system. Nevertheless, the number of CPN-based models concerning the verification of the pacemaker system is limited and insufficient.

Consequently, this study aimed to contribute to the verification and validation of the pacemaker system and to present a practical methodology when approaching similar systems. The model is formed of seven submodels to augment its legibility and adaptability. In addition, the model can assist not only in facilitating a better understanding of pacemakers and relevant cardiac events but also in producing customizable data to judge and optimize different relevant algorithms and techniques sufficiently.

Bibliography

[1] van der Aalst, W.M., Stahl, C.: Modeling business processes: a petri net-oriented approach. MIT press (2011)

[2] AIS Group: CPN Tools by The University of Technology, Eindhoven, The Netherlands. *www.cpntools.org* (2013)

[3] Assiri, M., Janicki, R.: Modelling and analyzing electrocardiogram events using timed coloured petri nets. In: Köhler-Bußmeier, M., Kindler, E., Rölke, H. (eds.) Proceedings of the International Workshop on Petri Nets and Software Engineering co-located with 41st International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2020), Paris, France, June 24, 2020 (due to COVID-19: virtual conference). CEUR Workshop Proceedings, vol. 2651, pp. 222–223. CEUR-WS.org (2020)

[4] Barbosa, P.E., Morais, M., Galdino, K., Andrade, M., Gomes, L., Moutinho, F., de Figueiredo, J.C.: Towards medical device behavioural validation using petri nets. In: Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on. pp. 4–10. IEEE (2013)

[5] Boston Scientific: Pacemaker system specification. Boston Scientific (2007)

[6] Boukredera, D., Maamri, R., Aknine, S.: A timed colored petri-net-based modeling for contract net protocol with temporal aspects. In: Proceedings of the Seventh International Multi-Conference on Computing in the Global Information Technology (ICCGI 2012). pp. 40–45 (2012)

[7] Bowden, F.D.: A brief survey and synthesis of the roles of time in petri nets. Mathematical and Computer Modelling **31**(10-12), 55–68 (2000)

[8] Gomes, A.O., Oliveira, M.V.M.: Formal specification of a cardiac pacing system. In: International Symposium on Formal Methods. pp. 692–707. Springer (2009)

[9] Hoare, C., Misra, J., Leavens, G.T., Shankar, N.: The verified software initiative: A manifesto. ACM Comput. Surv. **41**(4), 22–1 (2009)

[10] Iaizzo, P.A.: Handbook of cardiac anatomy, physiology, and devices. Springer Science & Business Media (2009)

[11] Jensen, K., Kristensen, L.M.: Coloured Petri Nets Modelling and Validation of Concurrent Systems. Springer, Berlin (2009)

[12] Jensen, K.: Coloured Petri Nets and the Invariant Method. Theoretical Computer Science **14**(3), 317–336 (1981)

[13] Jensen, K., Christensen, S., Kristensen, L.M.: CPN Tools State Space Manual. Department of Computer Science, Univeirisity of Aarhus (2006)

[14] K. Jensen: Coloured Petri Nets. Springer (1994)

[15] Knight, J.C.: Safety critical systems: challenges and directions. In: Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on. pp. 547–550. IEEE (2002)

[16] Macedo, H., Larsen, P., Fitzgerald, J.: Incremental development of a distributed real-time model of a cardiac pacing system using vdm. FM 2008: Formal Methods pp. 181–197 (2008)

[17] MacLeod, K.: An essential introduction to cardiac electrophysiology. World Scientific Publishing Company (2013)

[18] Majma, N., Babamir, S.M.: Medical software runtime checking using petri-nets & software agents. In: Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on. pp. 449–454. IEEE (2014)

[19] Majma, N., Babamir, S.M., Monadjemi, A.: Runtime verification of pacemaker using fuzzy logic and colored petri-nets. In: Fuzzy and Intelligent Systems (CFIS), 2015 4th Iranian Joint Congress on. pp. 1–5. IEEE (2015)

[20] Majma, N., Babamir, S.M., Monadjemi, A.: Utilizing fuzzy colored petri-nets to monitor cardiac pacemaker behavior. In: Application of Information and Communication Technologies (AICT), 2016 IEEE 10th International Conference on. pp. 1–5. IEEE (2016)

[21] Majma, N., Babamir, S.M., Monadjemi, A.: Runtime verification of pacemaker functionality using hierarchical fuzzy colored petri-nets. Journal of medical systems **41**(2), 27 (2017)

[22] Méry, D., Singh, N.K., et al.: Functional behavior of a cardiac pacing system. International Journal of Discrete Event Control Systems **1**(2), 129–149 (2011)

[23] Reisig, W.: Petri nets, an introduction, 2nd ed. Springer Berlin Heidelberg., Berlin (1991)

[24] Sharma, A.: Towards a verified cardiac pacemaker. Tech. rep., Technical Report NUS November (2010)

[25] Tuan, L.A., Zheng, M.C., Tho, Q.T.: Modeling and verification of safety critical systems: A case study on pacemaker. In: Secure Software Integration and Reliability Improvement (SSIRI), 2010 Fourth International Conference on. pp. 23–32. IEEE (2010)

[26] Westergaard, M.: CPN tools 4: multi-formalism and extensibility. In: Application and Theory of Petri Nets and Concurrency, pp. 400–409. Springer (2013)

[27] Woodcock, J.: First steps in the verified software grand challenge. Computer **39**(10), 57–64 (2006)

[28] Yang, S.: A petri net approach to remote diagnosis for failures of cardiac pacemakers. Quality and Reliability Engineering International **20**(8), 761–776 (2004)